REAL-TIME IMPLEMENTATION OF FACTORIZATION THEORETIC FEEDBACK CONTROLLER USING ADSP-2100

A Thesis submitted in
Partial fulfillment of the requirements
for the Degree of
Master of Technology

By

Vaishali Kulkarni

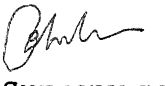
To the
Department of Electrical Engineering
Indian Institute of Technology-Kanpur

February-1994

CERTIFICATE

It is certified that the thesis entitled **"Real-time implementation of Factorization theoretic feedback controller using ADSP-2100"** by Vaishali Kulkarni, has been carried out under my supervision and that this work is not submitted elsewhere for a degree.

February-1994



Thesis Supervisor

Dr. V. R. Sule
Department of Electrical Engineering
I.I.T.-Kanpur.

Acknowledgements

I take this opportunity to express my deep sense of gratitude and sincere thanks to my thesis supervisor **Dr.V.R.Sule.** I am greatly indebted to him for the cooperation extended to me throughtout my work without which this thesis would not have taken this form.

I am thankful to Dr.Govind Sharma for allowing me to use the DSP Lab. I would like to thank Prasantan, Vishwanath, and all my friends for the help they extended during my work.

A special note of thanks is due to all my friends, who have contributed towards my enjoyable stay at I.I.T.K, a truly pleasant and memorable one.

Vaishali Kulkarni.

Dedicated To

My Parents

ABSTRACT

Factorization theory is a very powerful approach for multivariable control system design, with which the solutions to seemingly difficult and extremely important control synthesis problems can be characterized. One such important feature of this theory is the stabilization problem. The controller which would stabilize the plant is of the form C(F,R) where F is the fixed part depending on the plant and R is the free part which could be any stable transfer function with $||R||_\infty < 1$ for some closed loop performance in addition. The structure itself suggests a computational algorithm and could be implemented using presently available hardware. ADSP-2100 is a 16-bit processor from analog devices which is used for the implementation purpose. The aim is to test the practical feasibility of such feedback controllers.

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

SECTION 1.1 -- INTRODUCTION TO PROBLEM

1.1.1 A stabilization problem --

A control system forms a vital component of modern day technological processes like aircrafts, ships, chemical process plants, generators, amplifiers etc. A control system in most of such situations is required to automatically regulate small deviations of the process from its desired operating point.This is termed as the stabilization problem.

Stabilizing control laws in real life situations, have to be designed under ignorance or uncertainty of the model of the process. While the characterization of noise and disturbing environment are also not exactly known, the performance requirements of the system are also subject to changes with slight modification of the plant. In such situations feedback controllers, keeping in view the above uncertainties are necessary. In any such design of feedback controllers, a guarantee of closed looped stability with sufficient robustness to the plant variations is a foremost important issue. In this light some of the disadvantages of controllers being used presently are as follows.

(i) Practical implementation of well known PID and lead-lag type controllers require detailed parameter values for tuning of their parameters in order to guarantee closed loop stability. Thus with any step of tuning, the stabilization problem has to be resolved.

(ii) A more serious disadvantages of these standard controllers

is their preassigned structure (PID or lead-lag) which may not be the best suited for the plant and the given performance.

(iii) Conceptually, controllers designed using modern design theories are far superior in performance than the PID controllers.

1.1.2 Motivation for the thesis --

Although controllers of the type PID and lead-lag are simple to implement,this advantage is not very significant for modern day implementations using dedicated electronic hardware or using DSP algorithms. Modern means of controller implementations such as using dedicated DSP chips, ASIC based hardware, etc provide ways of implementing far more complex controllers than PID or lead-lag, even in real time.

1.1.2.1 Controllers based on factorization theory

In most practical situations the plant to be controlled can be considered as linear time invariant (LTI).For such models,factorization theory developed in recent times [1] offers powerful methods of synthesis of feedback systems.An important feature of this theory is the concept of " parametrization of controllers ".The basic problem can be stated as follows.

Consider the plant-controller set-up as shown in fig.1.1 . P represents a generalized plant which is assumed to be linear time invariant.C represents the controller. w is the set of disturbances,noise and reference signals; z the regulated variables of P; u the control input and y is the measured variables. The problem can be stated as follows.

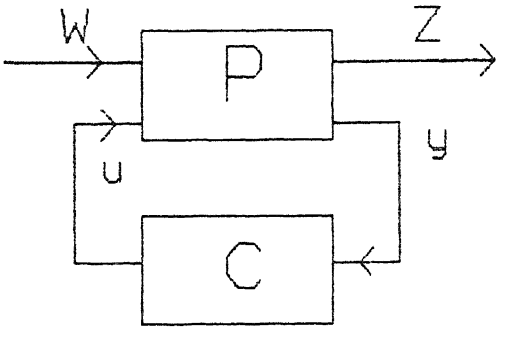


FIG. 1.1

PLANT-CONTROLLER SET-UP

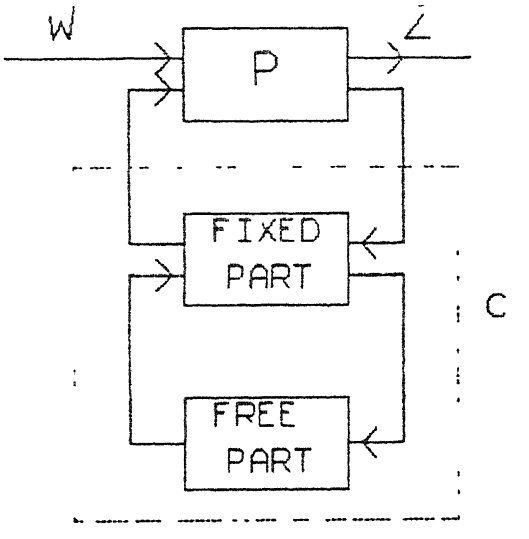


FIG. 1.2

CONTROLLER STRUCTURE

Given a LTI plant P, find all LTI controllers C such that

(i) C stabilizes P and (ii) The performance $||Hzw||_\infty < r$ where Hzw is the transfer function matrix between the exogenous input w and the regulated outputs z of the plant, and r is a specified constant, denoting an index of performance. The parametrization of all such controllers which solves the above problem is obtained in terms of fractions of stable transfer functions which are affine functions of a freely assignable affine part.

ADVANTAGES --

(i) One of the principle advantage offered in the above approach is the formulation of various design problems of feedback system as problems of approximation over linear spaces.

But from the viewpoint of controller implementation the next advantage, however seems to be most promising.

(ii) The controller which solves this problem ie which provide closed loop stability as well as the closed loop performance can be written in the form C(F,R) where F is a fixed data part decided by the plant model and R is a freely assignable stable transfer function. This controller formula can be implemented using a dedicated processor.

The controller structure along with the plant is shown in fig.1.2 .

The fixed part is determined by the plant model. Once this is known, the free part R can be chosen arbitrarily in a specified class. for example (i) R can be any stable transfer function for closed loop stability (ii) $||R||_\infty < 1$ in addition, for closed

loop performance.

SECTION 1.2 -- SCOPE OF THE THESIS

The structure of the factorization theoretic controllers discuued in the above section itself not only suggest a computational algorithm, but seems to be promising for implementation using some dedicated electronic hardware like signal processor chips. ADSP-2100 is a 16-bit processor from analog devices which is used for the implementation purpose. The aim in this thesis is to test the practical feasibility of such feedback controllers.The thesis in addition, also gives a brief summary of the available present technologies for industrial controllers.

SECTION 1.3 -- ORGANIZATION OF THESIS

Chapter 1 gives the introduction and motivation of the problem.

Chapter 2 discuss some of the present technologies and devices which could be employed for industrial controllers.

Chapter 3 gives the architectural features as well as the development tools of the ADSP-2100 which are used for designing the system.

Chapter 4 gives the actual design of the system.Both algorithmic and hardware aspects are discussed.

Chapter 5 concludes the thesis and shows directions for further work.

CHAPTER 2

DEVICES FOR REAL-TIME CONTROL

SECTION 2.1 -- COMPUTARIZED-CONTROLLED SYSTEMS

2.1.1 direct digital control --

The early installation of control computers operated in supervisory mode, either as operator guide or set-point control.Because the computers were so unreliable, they controlled the process by printing instructions to the process operator or by changing the set-points of the analog regulators.A drastic departure from this approach was made when analog technology was replaced by digital technology. Then came Direct Digital Control which means the computer controlled the process directly. [4]

In DDC most of the process control functions are performed by a control computer.These include data acquisition, both digital and analogue data conversion and formatting, computation, comparing with set-point limits, recording and monitoring of events etc.So basically the controller equipment reduces to transducer,actuators and the control computer.Rest other operations are performed by the digital computer itself.A schematic of DDC is shown in fig.2.1 .

Once a control equation is chosen to suit the dynamic characteristics of the process, the computer takes the control action.A control algorithm has to be prepared for that.Whenever the process parameters or the process requirements change,the control algorithm alone need to be altered and apparently can be done without any change in the hardware.The rate of change of the controlled varia-

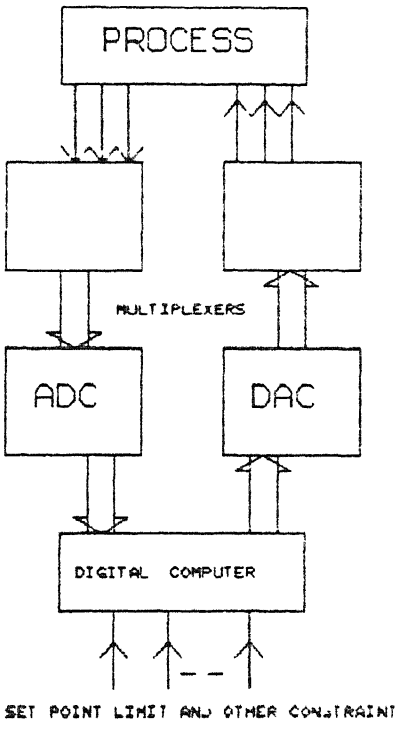


FIG. 2.1

DDC

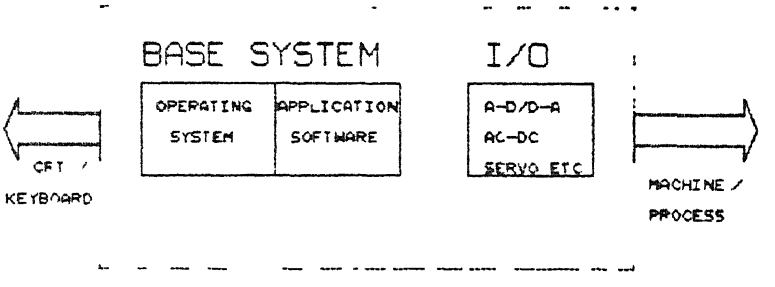


FIG. 2.2

PLC ARCHITECTURE

ble may be checked at any stage and altered by suitable programming schedule comparing with a given set of limiting values. This adaptability has made the DDC quite attractive.

2.1.2 Real Time implementation--

Since last 20 years computer-controlled system has evolved to a large extent, in a sense any type of system can be controlled using a computer.But as far as real-time operation is concerned, it poses a great challenge to both the software and hardware designer.

In a software controlled system,a package that carries all the activities for control is written which is to be executed by the computer.But in real-time operation all the software routines are controlled by the process. The actual time needed for executing depends on various subsystems which have different time delays,tolerances and complexities.Some of the problems that arise in real time are

(i) Computation schedule is governed by external process having several subsystems.Often the controller is supposed to control several devices and the problem becomes severe because of different time delays of each device.

(ii) Computation of some signal depend on the other subsystem and hence the response of that system may not be available at the time needed by the computer because of the delays involved.

(iii) Some systems are less tolerant and demand that every signal be delivered within a specified time only.

There are some real-time languages

like modula-2,Ada who meet most of the requirements of real time application.In practice most of the actions undertaken by the computer are initiated by software and then carried out by dedicated hardware.Sometimes certain tasks like arithmetic operation are carried out by a completely hardware unit to reduce the execution time.Thus computer-controlled systems are always a fine combination of both software and hardware aspects.

SECTION 2.2 -- MODERN DEVICES FOR CONTROL

2.2.1 Programmable Logic Controllers --

PLC's have emerged as one of the basic blocks in the industrial and process automation; replacing the more conventional relay systems. The basic disadvantages of the relay logic are

(i) Constraint on flexibility due to hard wiring type structure.

(ii) Frequent failure due to their mechanical operation.

PLC's due to their construction using solid state semiconductor technology offer an improved reliability.As the requirements of a particular control application change, the PLC's need only to be reprogrammed to change the logic diagram within the logic of PLC.One more reason for the success of PLC is the ease of man-machine interface.

The PLC architecture is shown in fig.2.2 . These are special purpose microcomputers.The architecture is divided into two parts; base system and I/O components.It has an operating system,usually an embedded one and some application software.PLC's are

programmed using software language known as "relay ladder".One important feature of this is that inputs/outputs control the process directly.Inputs from the plant and outputs of the plant are directly wired to the PLC system.Direct interfacing to the industrial devices of this sort often cause severe problems with computer control,because of electrical interference. But PLC's are designed to be inherently immune to these effect, being packaged to operate and be located in an industrial environment where there may be high noise,vibration,temperature or humidity.The PLC's has wider variety of industrially hardened I/O including ADC,DAC,serial data communication facilities and in certain cases medium performance servo control.

Modern developments in PLC's --

PLC's which were initially designed to replace relay racks were programmed using relay ladder logic and designed to operate sequentially to emulate relays.But new generation user's have no emotional ties with ladder logic and are forcing PLC makers to come up with alternative language such as C and Basic.Reliance automax's DCS is an example which can be programmed using higher level languages.Other industries like AllenBradley have come up with PLC's which employ open architecture wherein PLC's not only send information back to centralize mainframes but also to other controllers on the production line [5].

Considering again the cost of large PLC's,users are turning to smaller PLC's but having the I/Os of large ones.Such PLC's are also emerging the market.GE Fanuc,Siemens have developed such low cost PLC's which can handle even distributed I/O.

2.2.2 Digital Signal Processor Chips --

With the development of DSP chips,there has been a phenomenal growth in the signal processing activities from the view point of implementation.With the ready availability of cheap and reliable devices from Texas,Analog devices,Motorola as well with their support tools ,they offer an easy environment for real time systems.

The use of digital signal processor as a controller offer a number of potential advantages.In particular DSP's incorporate a high degree of parallelism and dedicated arithmetic circuits. This provide high resolution and high speed arithmetic. The architectures are often specifically oriented towards the implementation of appropriate algorithms, so the software development time can be reduced significantly and results in greater reliability.

Digital Signal processor is basically a microprocessor whose architecture is optimized to process sampled data at high rates.It performs such operations as accumulating the sum of multiple products much faster than an ordinary microprocessor can.The architecture of a generic signal processor chip is shown in fig.2.3 .

It's architecture is designed to exploit the repetitive nature of signal processing by pipelining the data flow for extra speed.A key feature of this chip is a fast array-multiplier and accumulator that allows multiply-accumulate operation to be executed in one clock cycle.The other features includes pipelining and parallelism, independent memories, 'circular and bit reversed addressing' modes.Most DSP's use the Harvard architecture,where instructions and data are kept in separate memories to allow the execution of several operations in parallel.In the architecture shown there are two memories X and Y. The arithmetic precision is greatly improved with the evolution of floating

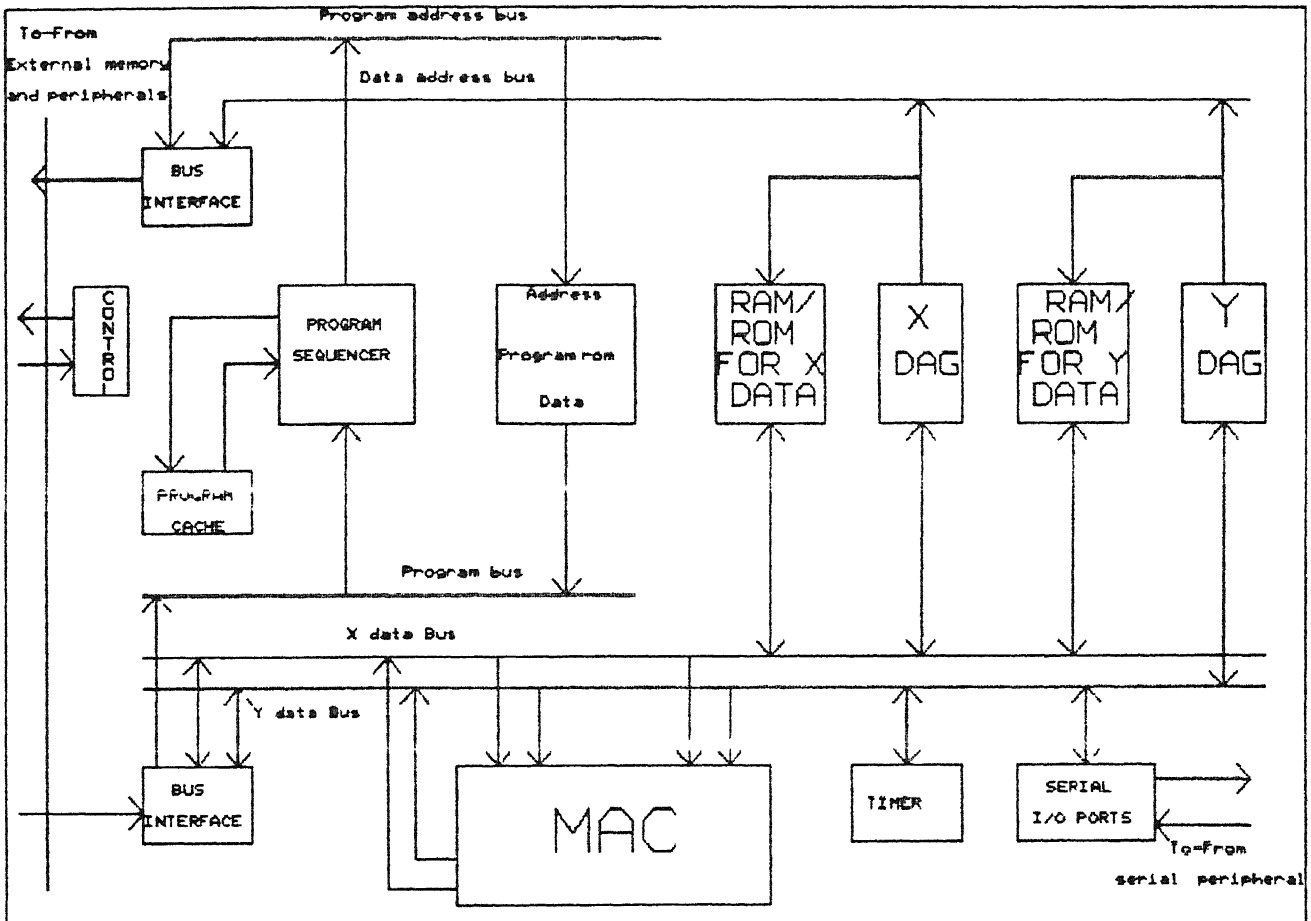


FIG. 2.3

GENERIC SIGNAL PROCESSOR CHIP

point processors.With the development of friendlier software and hardware tools,the programming and real-time operation has been simplified to great extent.

2.2.3 Microcontrollers --

Microcontrollers provide a mean of realizing a broad span of complex and yet flexible instruments.Microcontrollers were aimed at I/O intensive, yet algorithmically simple,real-time application.Evolved from the microprocessor system,the chip integrates memory,I/O devices and a chip level processor into a single unit.A hardware configuration of a typical microcontroller is shown in fig.2.4 .[9]

The key features include program execution from on chip ROM rather than external RAM and a general purpose microprocessor architecture.Data registers and general-purpose RAM are always partitioned from program store ROM.even though program and data memory are partitioned,program and data highways are often shared.

A variety of other hardware enhancements include sophisticated communication handlers and high speed I/O units.They are

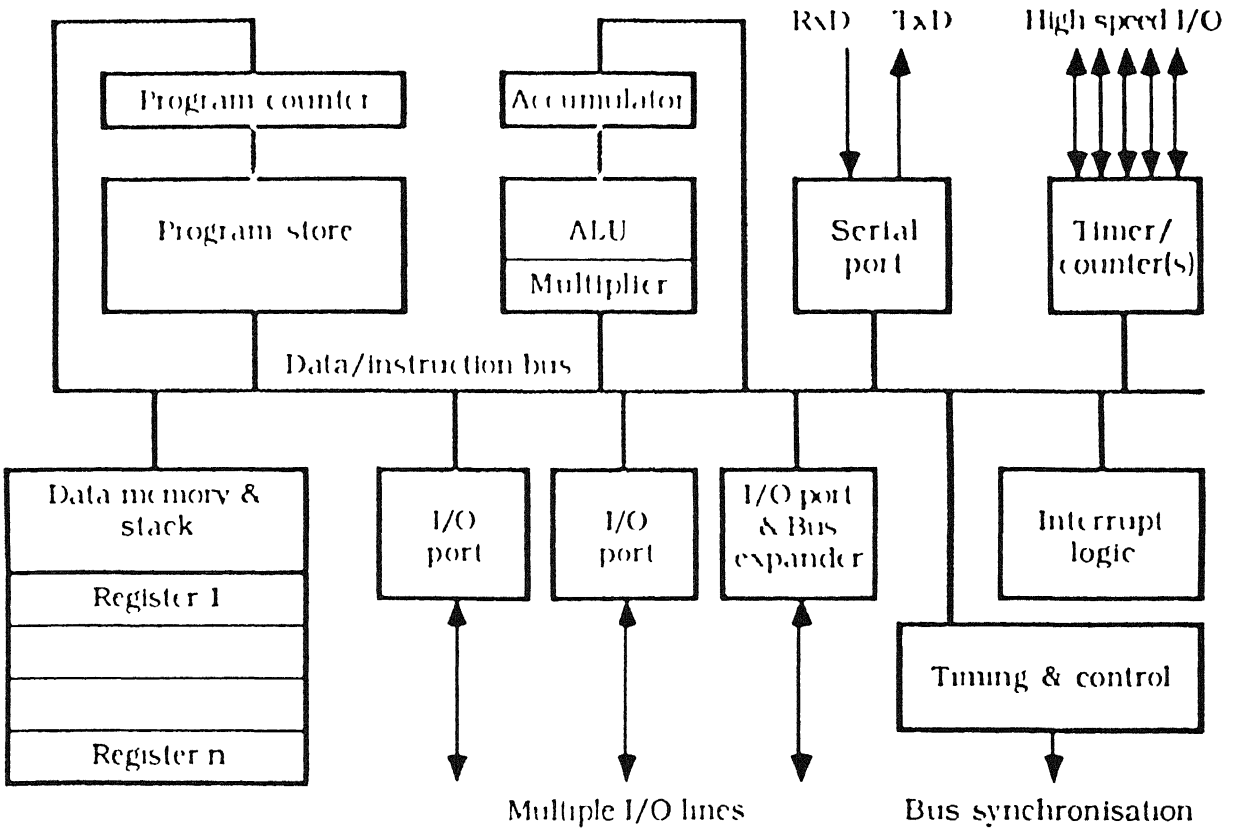
(i)Serial and parallel I/Os--

Always used as interfacing ports.Ports are bidirectional allowing them to be used as input as well as output.

(ii)Interrupt system --

Interrupts are always prioritized to two or more levels and interrupt masking is supported.

(iii)Timers/counters --



Simplified typical microcontroller hardware structure

FIG. 2.4

These may be configured to count either a divided version of system clock or transition on an input system.Watchdog timers are also provided as a means of restoring correct program flow in the event of hardware or software fault.

(iv)A-D converters --

Despite the considerable technical difficulties of integrating an analog functions with high noise susceptibility into digital systems, microcontrollers are integrated with data acquisition systems.

(v)Arithmetic Enhancements --

In addition to arithmetic circuits, a multiplier is also included.Program flow control is also made effective by adding program counter and index register increments.

Differentiation from signal processor --

DSPs were designed basically to handle signal processing tasks.There was no provision made for sophisticated multi-port I/O.Signal processing need flow data channels so a limited I/O capability therefore sufficed.Microcontrollers on the other hand were produced basically as means to replace hardwired logic so emphasis was placed more on flexible and multiple I/O facilities than on fast arithmetic.

Recent developments in microcontrollers --

Current application however show a trend towards a fast real-time control using substantial algorithms and multiple I/O requirements.Present microcontrollers have been designed with considerable hardware enhancements,particularly with respect to arithmetic capabilities and peripheral autonomy.So a new generation called as digital signal controllers has been introduced.

Announced by Texas TMS320C14 was the first device to combine the functionality of microcontrollers with the performance of DSPs.Being a first generation part,there are certain restrictions which arise as a result of its direct evolution from TMS320C1X family.They are associated with the limited I/O channel count of TMS and with the unsophisticated Interrupt system.

So it is likely that the future would see further convergence of digital signal controllers and microcontrollers.Next generation signal controllers are likely to offer peripheral handling,bit manipulations and interrupt implementation comparable to that of microcontrollers.

2.2.4 ASIC based Mixed Signal system --

From the point of view of real-time system,development of mixed signal system on a chip is of direct relevance.This a novel phenomena which provides means of mixing analogue and digital functions on a chip.Examples are Brooktree RAMDAC,national Semiconductor etc.Fig.2.5 shows a configuration of a typical mixed signal IC. [9]

Fig.2.5 shows the close tie up between the analogue section and the conventional microprocessor. These new devices are assumed to contain over 20,000 mixed signals.They offer challenging opportunities to system design, both in terms of on-chip design and their utilization in practical engineering problems.

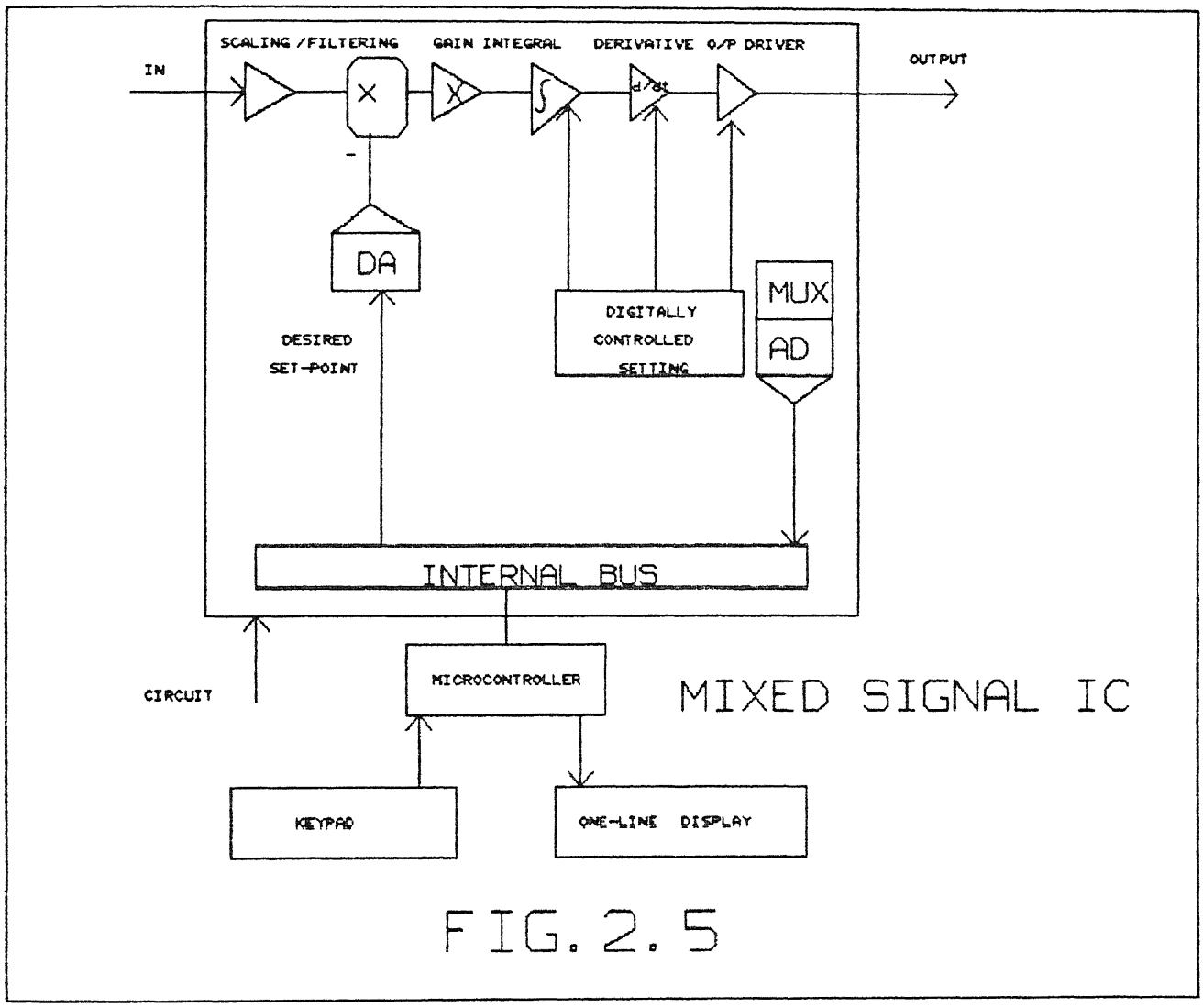

SCALING /FILTERING
GAIN INTEGRAL
DERIVATIVE O/P DRIVER
IN
OUTPUT
X
X
∫
d/dt
-
DA
DESIRED
SET-POINT
DIGITALLY
CONTROLLED
SETTING
MUX
AD
INTERNAL BUS
CIRCUIT
MICROCONTROLLER
MIXED SIGNAL IC
KEYPAD
ONE-LINE DISPLAY

FIG. 2.5

CHAPTER 3

ADSP-2100 SIGNAL PROCESSOR

SECTION 3.1 -- ADSP-2100

Most of the real-time problems of computation are difficult to implement in real time using general purpose computers.To solve these problems two different approaches have traditionally been followed:
(i) To use design specific hardware architectures or (ii) To use powerful general computers like data-flow machines,systolic arrays etc with high degree of parallelism and pipeline structured modes.

However in recent years alternatives have on the market of which ADSP-2100 is an example. This family combines the flexibility of a high speed controller with the numerical capability of an array processor offering an inexpensive alternative.

3.1.1 General description and features --

The ADSP-2100 is a programmable single chip microprocessor optimized for digital signal processing and other high numeric processing application.To date most DSP chips dedicate a large portion of their silicon area to the on-chip memory which not only constraints the size of the memory but also con fines the processing logic to a smaller area and reduces the functionality.The 2100 uses an alternate approach to access external memory efficiently.with the exception of a small instruction cache, the chip contains itself

no memory.The enormous amount of the memory is used to add functionality and increase processing throughput significantly.

The ADSP-2100 includes items such as a full-function barrel shifter for normalization and denormalization,two independent data address generators with modulo addressing capability,a program sequencer with provision for zero overhead looping,a background register set for rapid context switching,and sufficient internal busing to support a high degree of parallelism in the instruction set.An advanced 1.5-micrometer CMOS process gives the chip an instruction cycle time of 125 nsec and power consumption of less than 1/2 watt.

3.2.1 System configuration --

Fig.3.1 shows the basic configuration of the ADSP system.The processor interfaces with two external memory system,a program memory and data memory. Because they are separate,instruction and data can be accessed simultaneously.Data is also allowed in the program memory which also allows dual data access.

A set of address.data and control lines is also provided for each memory.on the program memory side there are 14 address lines (PMA),24 data lines(PMD),a memory select signal(PMS),read and write strobes(PMRD and PMWR),and a signal to indicate when data (as opposed to an instruction) is being accessed (PMDA).The 14 address lines give an address range of 16K words,which can be expanded to 32K if the PMDA signal is used as an additional

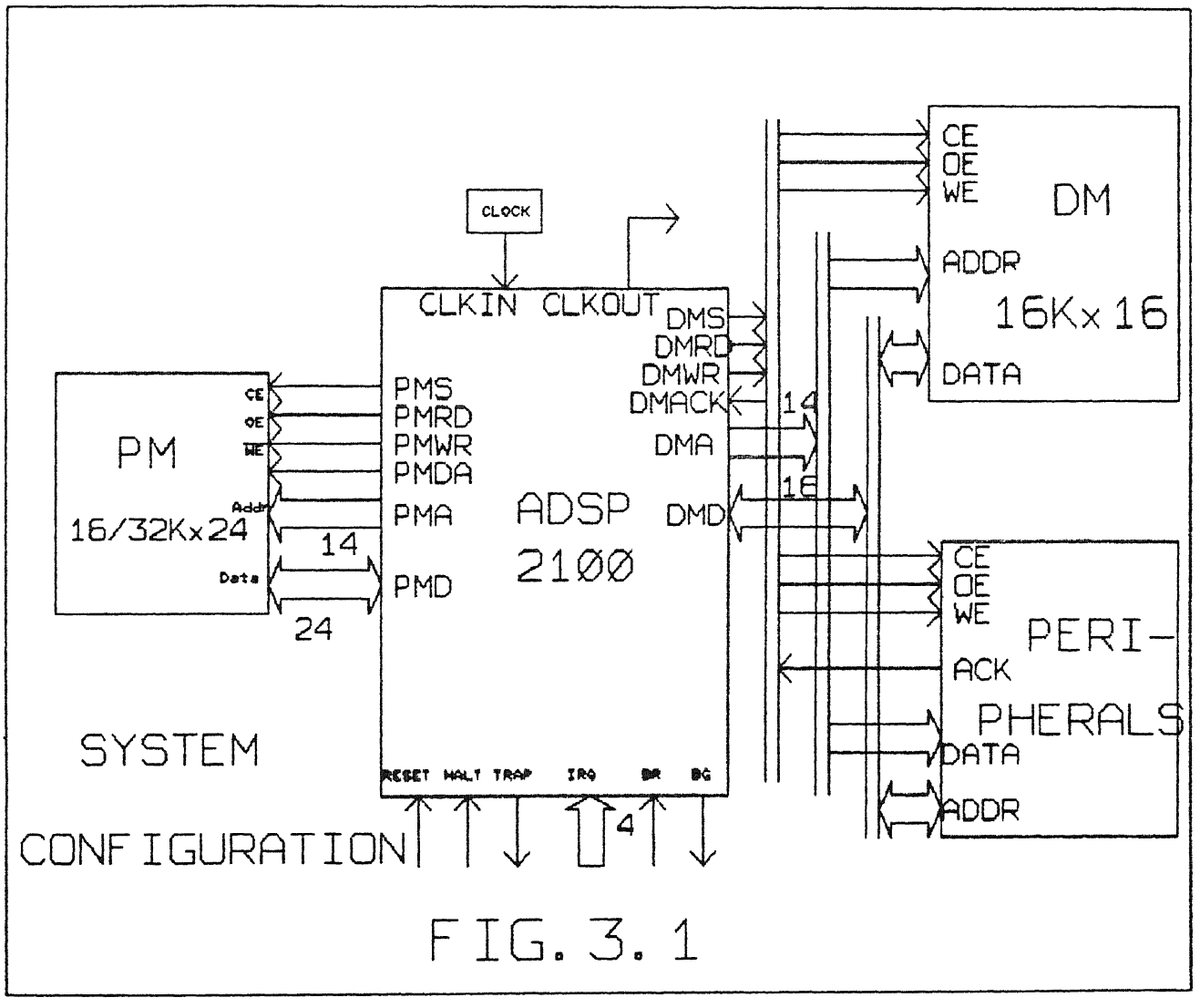

CLOCK
CLKIN
CLKOUT
ADSP
2100
PMS
PMRD
PMWR
PMDA
PMA
PMD
DMS
DMRD
DMWR
DMACK
DMA
DMD
14
16
24
4
RESET
HALT
TRAP
IRQ
BR
BG
PM
16/32Kx24
CE
OE
WE
Addr
Data
DM
16Kx16
ADDR
DATA
PERI-
PHERALS
ACK
SYSTEM
CONFIGURATION

FIG. 3. 1

address bit.

On the data memory side there are 14 address lines (DMA),16 data lines(DMD),a memory select (DMS),read and write strobes (DMRD and DMWR),and a signal to acknowledge the transfer of data (DMACK). Peripheral devices are memory mapped into the data memory address space.Slower devices can stretch the memory cycle as needed by withholding the DMACK signal.

The chip supports multiprocessing application with bus-request and bus-grant signals (BR and BG).The 2100 responds to a bus request by halting program execution and releasing the address, data,and control lines to the memories so that another processor can access them directly.Four interrupt request (IRQ) inputs are provided external devices that need periodic service from the processor.

The interrupt pins can be individually programmed for either level or edge sensitivity.The four inputs are prioritized with options for nesting (higher priority levels interrupting lower ones)or blocking (only one level serviced at a time). The maximum response time for an unmasked interrupt request is two cycles.

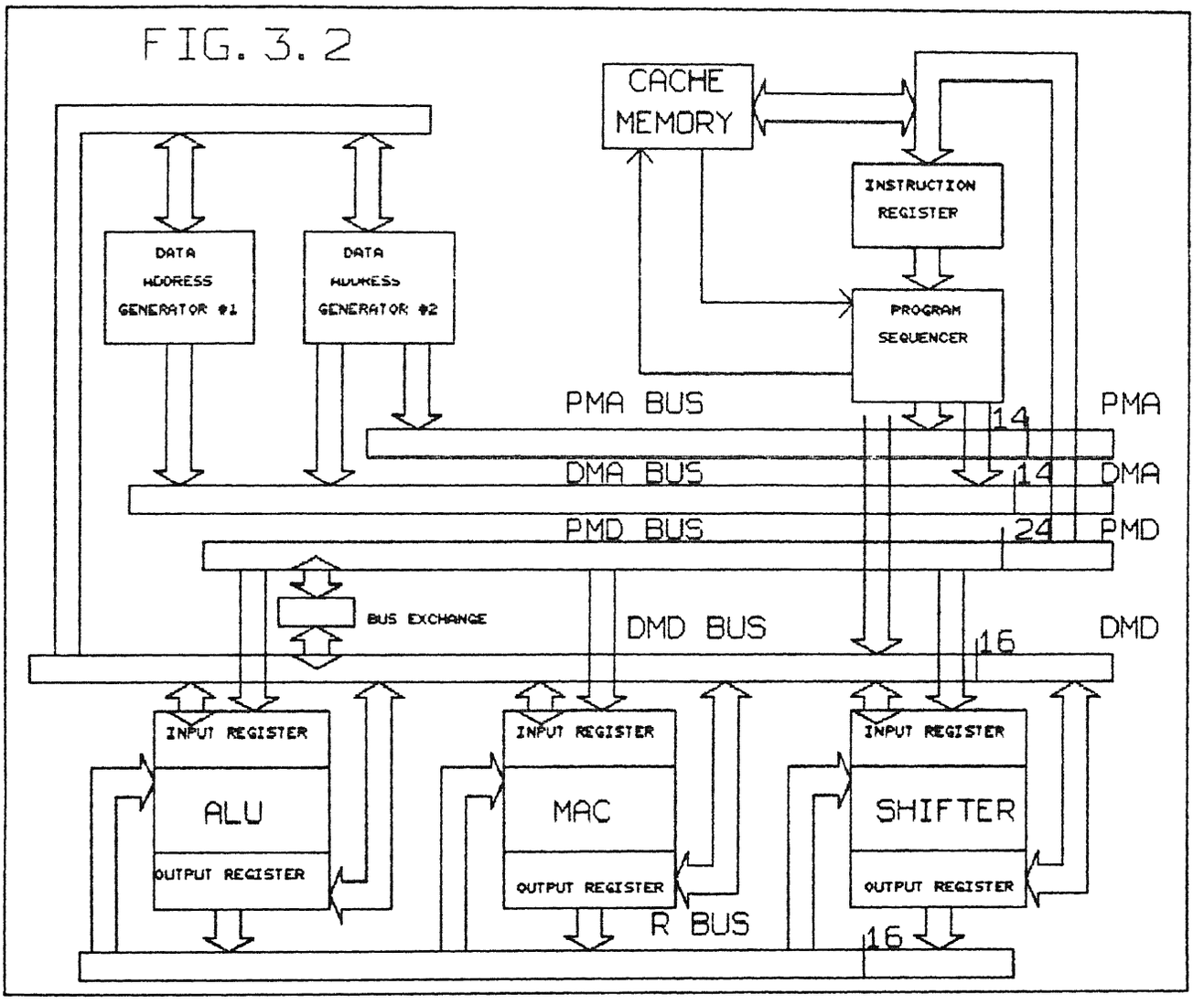
SECTION 3.2 -- ARCHITECTURE OVERVIEW

The internal architecture of the processor is shown in fig.3.2 .The overview of the internal architecture is giv here.All the components shown are supported by five internal buses.[6]

(i)Program memory address bus(PMA)

(ii)Program memory data bus (PMD)

FIG. 3.2


CACHE MEMORY
INSTRUCTION REGISTER
PROGRAM SEQUENCER
DATA ADDRESS GENERATOR #1
DATA ADDRESS GENERATOR #2
PMA BUS
14
PMA
DMA BUS
14
DMA
PMD BUS
24
PMD
BUS EXCHANGE
DMD BUS
16
DMD
INPUT REGISTER
ALU
OUTPUT REGISTER
INPUT REGISTER
MAC
OUTPUT REGISTER
INPUT REGISTER
SHIFTER
OUTPUT REGISTER
R BUS
16

(iii) Data memory address bus (DMA)

(iv) Data memory data bus (DMD)

(v) Result(R) bus(which interconnects all the computational units)

The PMD bus serves to transfer the instructions from off chip memory to the internal register which are executed and the next instructions are fetched thus introduces one level of pipelining in the program flow.This bus is 24 bits wide to accommodate the 24 bit instruction width and can also be used to transfer data to and from computational units through direct path or via PMD-DMD bus exchange unit.The PMA bus is 14 bits wide allowing direct access of upto 16K words of instruction code and 16K words of data.The DMD bus is 16 bit wide and provides path for the contents of any register in the processor to be transferred to any other register or to the external memory.The DMA bus is 14-bits wide allowing access of upto 16k words of data memory.

3.2.1 Computational units --

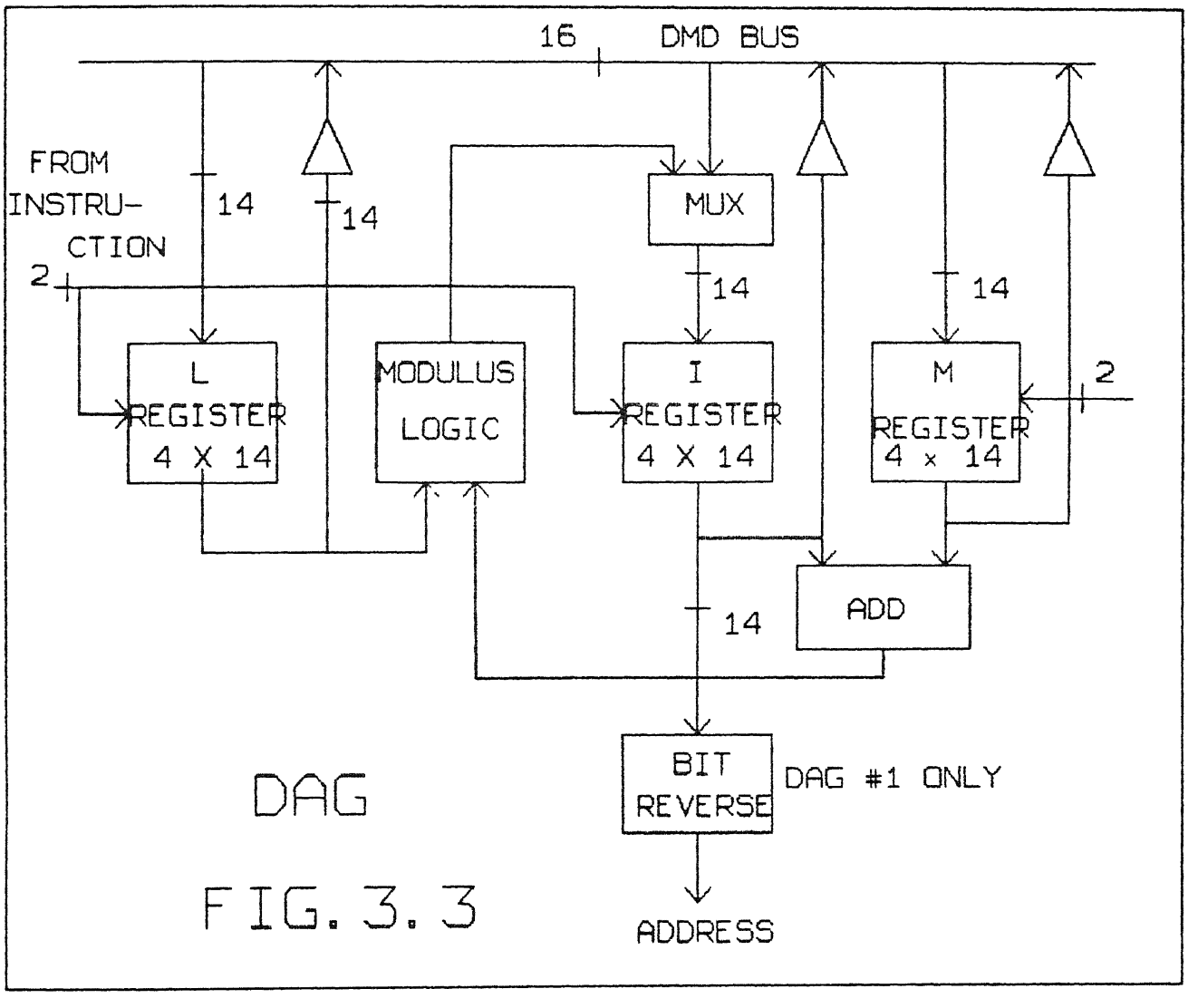
The computational units of the processor is divided into three independent units.Rather than being arranged in the usual series these units are rest side by side,relying on the R bus as a interconnect path.Operation of the R bus allows any sequence of arithmetic operation to be performed smoothly,without excessive juggling of the intermediate results.The ADSP-2100 contains three full-function and independent computational units:The arithmetic and logic unit, a multiplier/accumulator and a barrel shifter.

The computational units process 16-bit data with options such as saturation,multipricision and unbiased rounding of the final result. .All computational units contain a set of dedicated input and output registers.All these units in addition have a complete set of background registers

A set of background input and output register can be activated at any time if the processor must change tasks quickly.This capability effectively doubles the number of the available registers and can eliminate the save and restore overhead associated with the context switching.So for example,the execution of the interrupt service routine that require the computational facilities of the processor can be sped up tremendously.By switching to the background register set, the processor can save its current computational state in one cycle.Switching back to the original register will then restore the previous context at a later time.

3.2.2 Data address generators (DAGs) --

The ADSP-2100 contains two independent data address generators so that both data and program memory can be accessed simultaneously (fig.3.3).The DAGs provide indirect addressing capabilities.Both perform automatic address modification.The two DAGs differ In a few respects. DAG1 can only generate data memory addresses,but provides an optional bit-reversal capability.DAG2 can generate both data memory addresses,but has no bit-reversal capability.Thus the processor has the capability of fetching the two simultaneously,one from data memory and one from program memory.


16
DMD BUS
FROM
INSTRU-
CTION
2
14
14
MUX
14
14
L
REGISTER
4 X 14
MODULUS
LOGIC
I
REGISTER
4 X 14
M
REGISTER
4 x 14
2
14
ADD
BIT
REVERSE
DAG #1 ONLY
ADDRESS

DAG

FIG. 3.3

Each data address generator contains the registers as follows.Memory pointers are kept in the I(index) register file.The M(modify) register file contains incremental values,which move the the pointers by the desired amount each time they are used.The L(length) register define the size of the data structure being accessed.Each of the register file contains 14-bit registers,which are loadable and readable from the internal DMD bus.

In addition to the direct addressing and indirect addressing the address generators support modulo addressing as well as bit reverse addressing The modulo addressing logic implements automatic wraparound for circular buffers i.e. if the sum of the M register and the I register would cross the boundary of the buffer,the modified I register must wrap around .The bit reverse addressing logic is primarily used in FFT computation in order to aid the scrambling or unscrambling of the data.

Whenever an indexed memory reference is made,a I register provides the address.An independently selected M register is then added to the address to form the tentative address.The tentative next address feeds into the modulus logic along with the selected L register value.The modulus logic determines whether the new address is outside the bounds of the data structure.If it is,the address wraps around in the modulo fashion to remain within the allowable range. Otherwise the address passes through the modulus logic unchanged.In either the output of the modulus logic is loaded back into the original I register,ready or the next memory reference.The complete address modification process can be described with the

following formula:

Next address=(I+M-B)modulo(L)+B

I=Index register value

M=Modify register value

B=Base address

L=Length register value

3.2.3 Program sequencer --

Keeping numerical throughput high also requires a sophisticated program sequencer,for if the processor gets bogged down by branching,looping,or responding to interrupts,the computational rate suffers.A large portion of the 2100 chip was dedicated to the program sequencer to streamline the program flow and minimize overhead (fig.3.4).

Instruction address can come from four possible sources:a 14-bit program counter(PC),an internal 16-level pc stack,an interrupt controller,or a 14-bit field of the instruction register.The program counter keeps track of the current instruction address and feeds the incrementer,which provides the next contiguous address.The pc stack stores the subroutine and the interrupt-return addresses and is chosen when returning to the main program execution.The interrupt controller monitors the external interrupt-request inputs and provides jump vectors when needed.The instruction registers is chosen when a direct jump is executed.

The 2100 includes status registers to keep track of the arithmetic results,execution modes,and interrupt con-

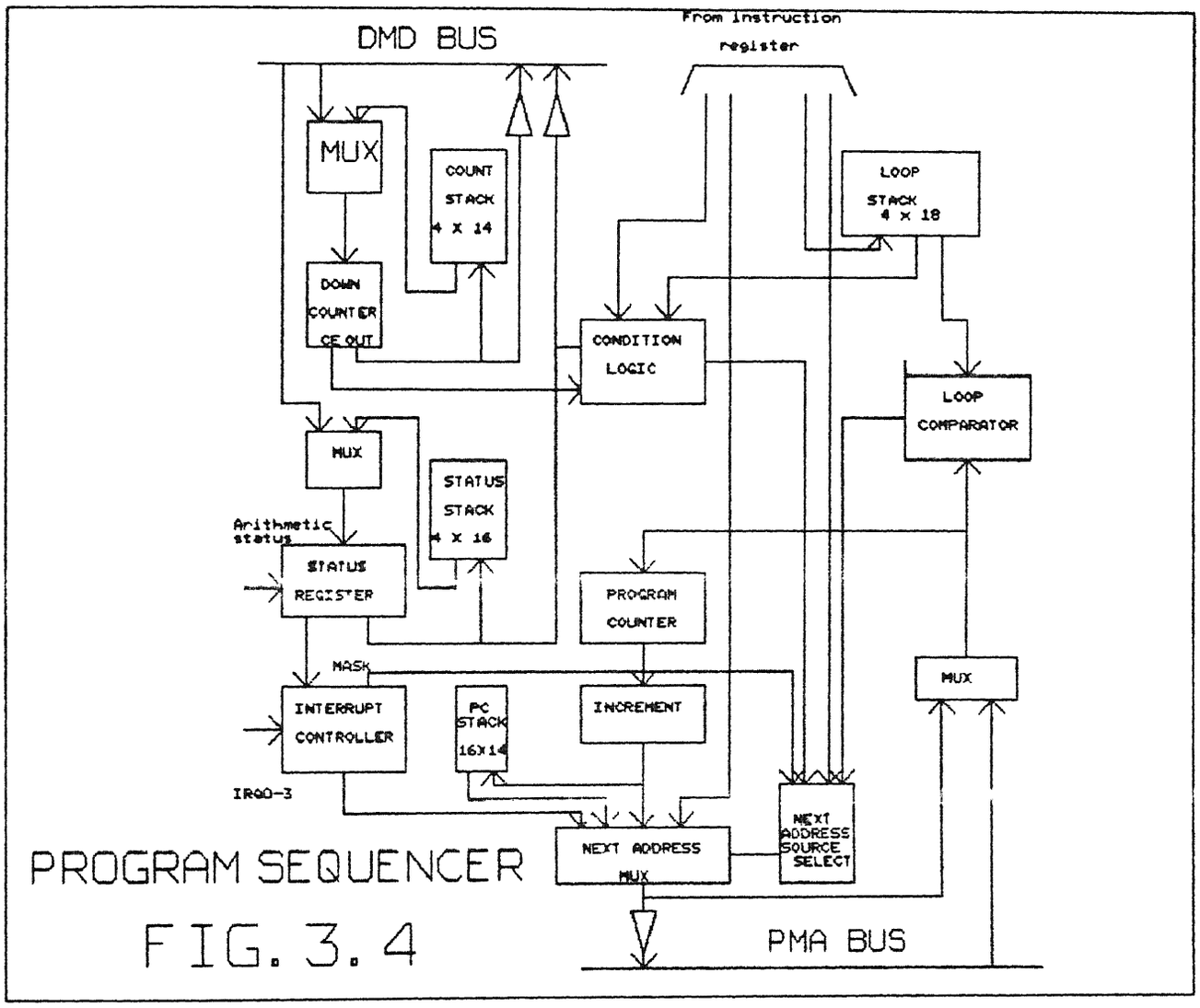

DMD BUS
From instruction register
MUX
COUNT STACK 4 X 14
LOOP STACK 4 X 18
DOWN COUNTER CE OUT
CONDITION LOGIC
LOOP COMPARATOR
MUX
STATUS STACK 4 X 16
Arithmetic status
STATUS REGISTER
PROGRAM COUNTER
MASK
MUX
INTERRUPT CONTROLLER
PC STACK 16X14
INCREMENT
IRQ0-3
NEXT ADDRESS SOURCE SELECT
NEXT ADDRESS MUX
PROGRAM SEQUENCER
PMA BUS

FIG. 3.4

figuration.Arithmetic status dives the condition logic that controls the selection of a next address for conditional operations.Interrupt configuration status transfers to the interrupt controller.An internal four-level-deep stack saves status information automatically when vectoring to an interrupt service routine and restores it upon return.The status stack can be pushed or popped manually at any time.

The down counter controls program looping with the decrement and branch features.Preloaded via the internal bus DMD,it generates a counter-expired status (CE) output when the count reaches zero.Decrementing occurs automatically every time the status is checked.A four-level stack associated with the counter allows counted loops to be nested five levels deep.

The loop-stack and the comparator also facilitate program looping.The do-until instruction sets up these functions.When executed,this instruction pushes the end-of-loop address and termination condition onto the stack and the beginning-of-loop address(PC+1) onto the PC stack.Once the loop is entered,the loop comparators compares the next address output of the sequencer with the end-of-loop address on the loop stack.When the two are equal,it indicates that the processor is fetching the last instruction in the loop.During the next cycle if the termination condition is false the PC stack is chosen as the next address otherwise the sequencer exits the loop by using the PC+1 as the next address. This automatic looping mechanism eliminates the need for an explicit jump instruction within the loop.

3.2.4 Cache memory --

The instruction Cache memory stores a short history of upto sixteen previously executed instructions.When the next instruction is already contained in the cache the cache direct-feeds the instruction register,freeing up the PMD bus for data transfers.The operation of the cache follows this scenario.

(i) In the write mode each instruction is also written in the cache.

(ii) When the PMD bus is busy with a data transfer,the instruction register is loaded from the cache.

(iii) If the loaded instruction is valid it will be executed on the next cycle.If it is not the instruction register is cleared.An additional cycle is now required to fetch the next instruction.The validity is determined by the cache monitor.

The cache is an important part of the efficient utilization of the ADSP-2100.When the data can be read from the program memory the ADSP-2100 in effect becomes a processor with two data buses.Cache operation is completely transparent.No maintenance or overhead is required for either the storage or use of instructions in the memory.

3.2.5 PMD-DMD Bus exchange --

The PMD-DMD bus exchange unit couples the program memory data bus to the data memory data bus,allowing them to transfer in both the directions.Since the PMD bus is 24-bit wide,while the DMD bus is 16-bit wide,only the upper 16 bits

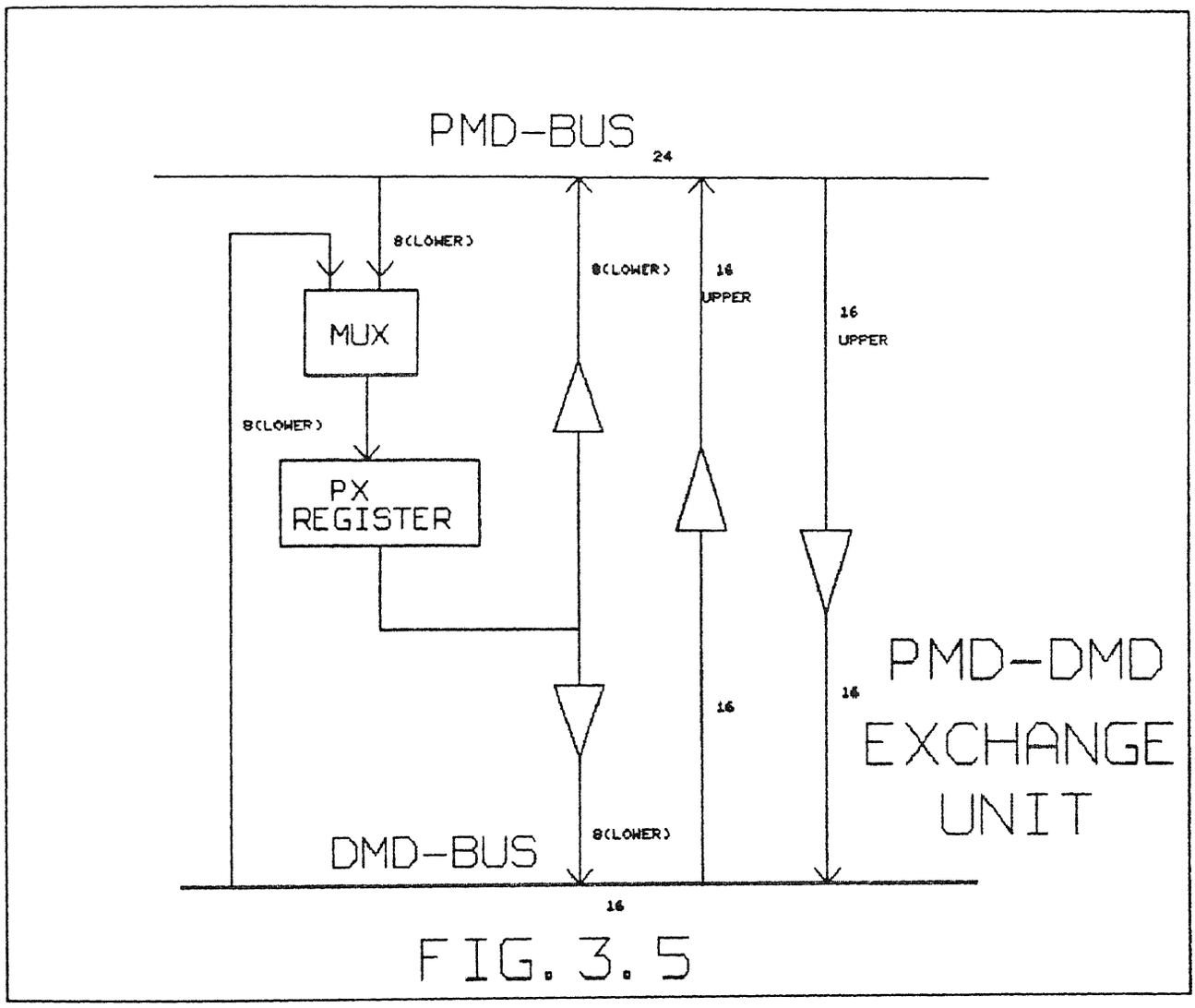

PMD-BUS
24
8(LOWER)
8(LOWER)
16
UPPER
16
UPPER
MUX
8(LOWER)
PX
REGISTER
PMD-DMD
EXCHANGE
UNIT
16
16
DMD-BUS
8(LOWER)
16

FIG. 3.5

can be directly transferred.An internal register (PX) contains the additional 8-bits.The register can be directly read and loaded when the full 24-bits are important.

SECTION 3.3 -- DEVELOPMENT SYSTEM

The ADSP-2100 development system is a complete set of development tools for systems using the ADSP-2100 DSP microprocessor. These tools are valuable in implementing an ADSP-2100 system design.With the cross-software system, the user defines the target system, writes the program, assembles each program module, links all modules to form a running system, simulates execution of codes etc.The ADSP-2100 evaluation board is an additional tool for evaluating the processor in real-time.The board also provides analog I/O and thus aids in the development process.[3]

3.3.1 Software development tools

The Cross-Software Development System includes these modules:

(i) System Builder-->

This module allows the designer to specify the amount of RAM and memory-mapped I/O ports for the target hardware environment.It uses high-level constructs to simplify this task.This specification is used by the other modules in the Cross-Software Development System.

(ii) Assembler -->

This module assembles your source code and data modules.It supports the high-level syntax of the instruction set.To support modular code development,the Assembler provides flexible macro processing and condi-

tional assembly.

(iii) Linker -->

The linker links the separately assembled modules.It maps the linked code and data output to the target system hardware,as specified by the System Builder output.

(iv) Simulator -->

Simulation is a process by which the characteristics and conditions of the real-world system or device are either mathematically modeled or reproduced. This allows the performance of the system or device to be investigated and experiments carried out independently of the actual system or device.

A DSP software simulator is a software package that fully models the internal registers,memory map,instruction set and operation of specific digital signal processor and verifies the program in non-real time. This module preforms an instruction level simulation. The simulator fully simulates the hardware configuration described by the system builder module.It flags illegal operation and several displays of the internal operation of the ADSP-2100.

The typical development cycle is shown in fig.3.6. The assembly language code is first written and assembled.If the software consists of multiple modules they must be linked together.This linked object code can then be loaded into the simulator.The software simulator is a powerful tool in the software development process,which allows code to be executed and bugs identified and corrected quickly and effectively.Once debugging is complete,the linked object code can be either downloaded directly to the target hardware,or loaded into in-circuit emulator for checking and

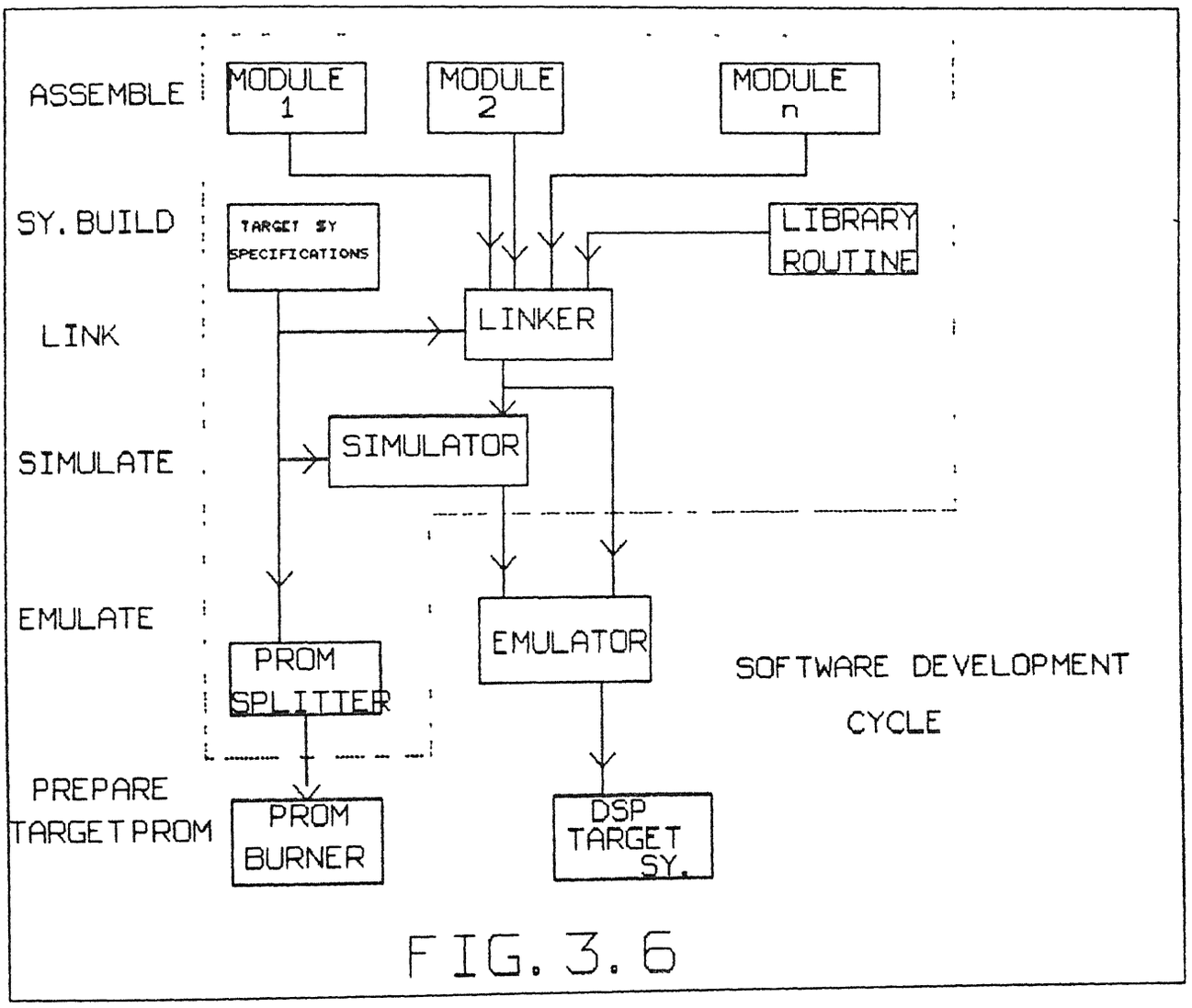

ASSEMBLE
MODULE 1
MODULE 2
MODULE n
SY. BUILD
TARGET SY SPECIFICATIONS
LIBRARY ROUTINE
LINK
LINKER
SIMULATE
SIMULATOR
EMULATE
EMULATOR
PROM SPLITTER
SOFTWARE DEVELOPMENT CYCLE
PREPARE TARGET PROM
PROM BURNER
DSP TARGET SY.

FIG. 3.6

monitoring the integration of the target hardware and software.

For a software simulator to effectively execute a program,there must be facilities available for connecting the simulated processor to the 'outside world'.To simulate I/O channels to the processor, they are assigned to the data files, enabling 'real' input data to be used and output data to be collected for examination and analysis.

During program execution,the internal registers and memory locations of the simulated processor are updated as each instruction is interpreted.Breakpoints can be established,triggered on read or write instruction to the specific memory locations.Execution is suspended when a breakpoint is reached.Once execution is suspended,registers and memory locations can be inspected and/or modified and simulation restarted.

In summary the main features of simulator are:

* Data files associated with I/O channels.
* Breakpoints programmable to trigger on:
  - Memory reads or writes
  - instruction acquisition
  - Specific data patterns
  - Error conditions
* Timing analysis in terms of clock pulses
* Interrupt generation at user-defined intervals.
* Trace facilities
* Display and modification of memory contents.
*Inspection and modification of registers.

3.3.2 The evaluation board

The ADSP-2100 Evaluation board is used for evaluating the ADSP based system in real-time.Evaluation board consist of ADSP-2100, 2Kx16 of data memory, 2Kx24 of program memory instruction space. A bidirectional coder/decoder channel and an undedicated 12-bit linear digital-analog converter.In addition,four BNC connectors are available for interfacing to external instrumentation.The Evaluation board runs under the control of on-board host processor enabling to access a variety of powerful debugging tools.The block diagram of the Evaluation board is shown in fig.3.7 .[7]

(i)memories--

The Evaluation board has two memories.One 2Kx24 program memory for code as well as 2Kx16 for program data and 2Kx16 data memory.The board peripherals are memory-mapped into this 16K address space in 2K block intervals.The 2K blocks of data memory space are enabled by the outputs of a 1-8 decoder.The ADSP's three most significant data memory address bits are decoded to eight enable signals which can be used for external peripherals.The board has a facility for extending the memories to full 16K.

(ii) Digital-Analog Converter--

DAC is one of the memory-mapped peripheral of ADSP-2100 that can be written to using the data memory write commands.The DAC provided is intended for use as an analog output for the display of processed data e.g.on an oscilloscope.It is not intended as a mean of reconstructing sampled data.The DAC accepts data from the ADSP at the processor full speed plus an additional wait cycle.The DAC has a settling time of 4 microsec.and thus cannot be written to faster than every 4 microsec.

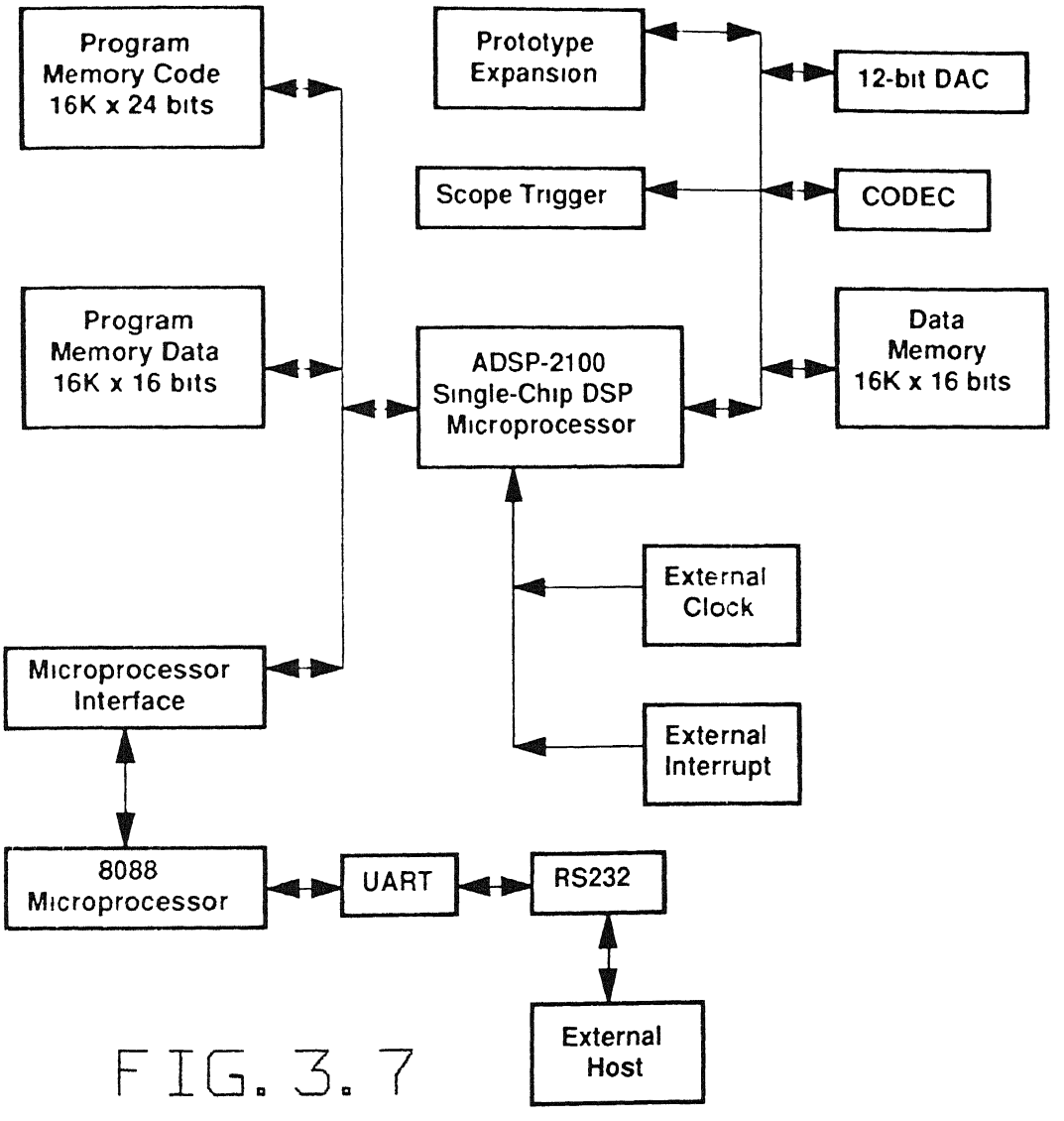


FIG. 3.7

ADSP-2100A Evaluation Board Block Diagram

(iii) ADSP-Host interfacing unit--

Interfacing unit is a mean for communication between the processor and the external host through which user can communicate with all his data.The 8088 microprocessor acts as a transfer device.The 8088 is defined as a memory-mapped port which has access to both the program memory and data memory.After assembling and linking the user can transfer all his data and program for real-time operation.The transfer takes place in serial mode.

The RS-232 serial connector are used for host and terminal communication.The personal computer is connected with VT100 emulation software. Many programs provide VT100 terminal emulation for IBM PC which supports the following

(a) Complete VT100 escape sequence emulation.

(b) 9600 Baud,no parity,one stop bit,XON/XOFF protocol.

(c) Character pacing and/or line pacing or echo wait mode.

(d) XON/XOFF flow control

PROCOMM (2.3 Version) is an IBM PC terminal emulation package with these necessary features.The board supports asynchronous XON/XOFF protocol.

CHAPTER 4

SYSTEM DESIGN

SECTION 4.1 -- ALGORITHM FOR FACTORIZATION CONTROLLER

4.1.1 Controller-observer structure --

The controller formula for the stabilizing controller C is developed in the Appendix A (eqn.2). This uses the Free part R at two places causing an overhead in terms of computational time.The observer-controller structure configuration [1] described below uses R at only one place so the overall computational time is reduced.

Let P be the discrete LTI plant with r.c.f (see Appendix A) Np and Dp. The input and output relation is given by

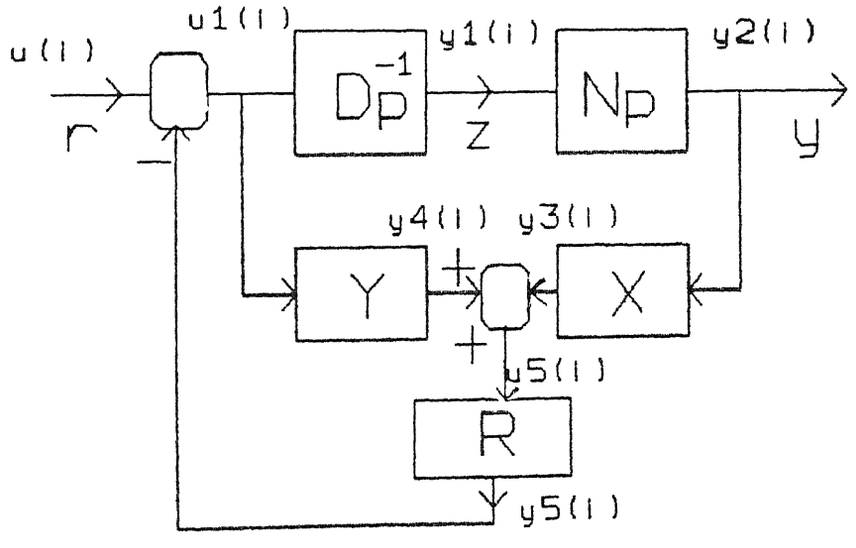
$$y = NpDp^{-1} u$$

Let z be any partial state available.(fig.4.1) Corresponding to the r.c.f of the plant, there exist X and Y which satisfy YDp+XNp=I. So using this, the observer shown in fig.4.1 reconstructs the partial state z. The controller part ie R then feeds back this state z.

4.1.2 Software structure --

The overall software structure is shown in fig.4.2 .

The ADSP-2100 interfaces with peripherals as shown ie with ADC/DAC plus a host terminal.The discrete LTI plant is assumed to be modeled in state-space form.The controller algorithm involves fractional representation of the plant, the state-space realization of



CONTROLLER-OBSERVER

STRUCTURE

FIG. 4. 1

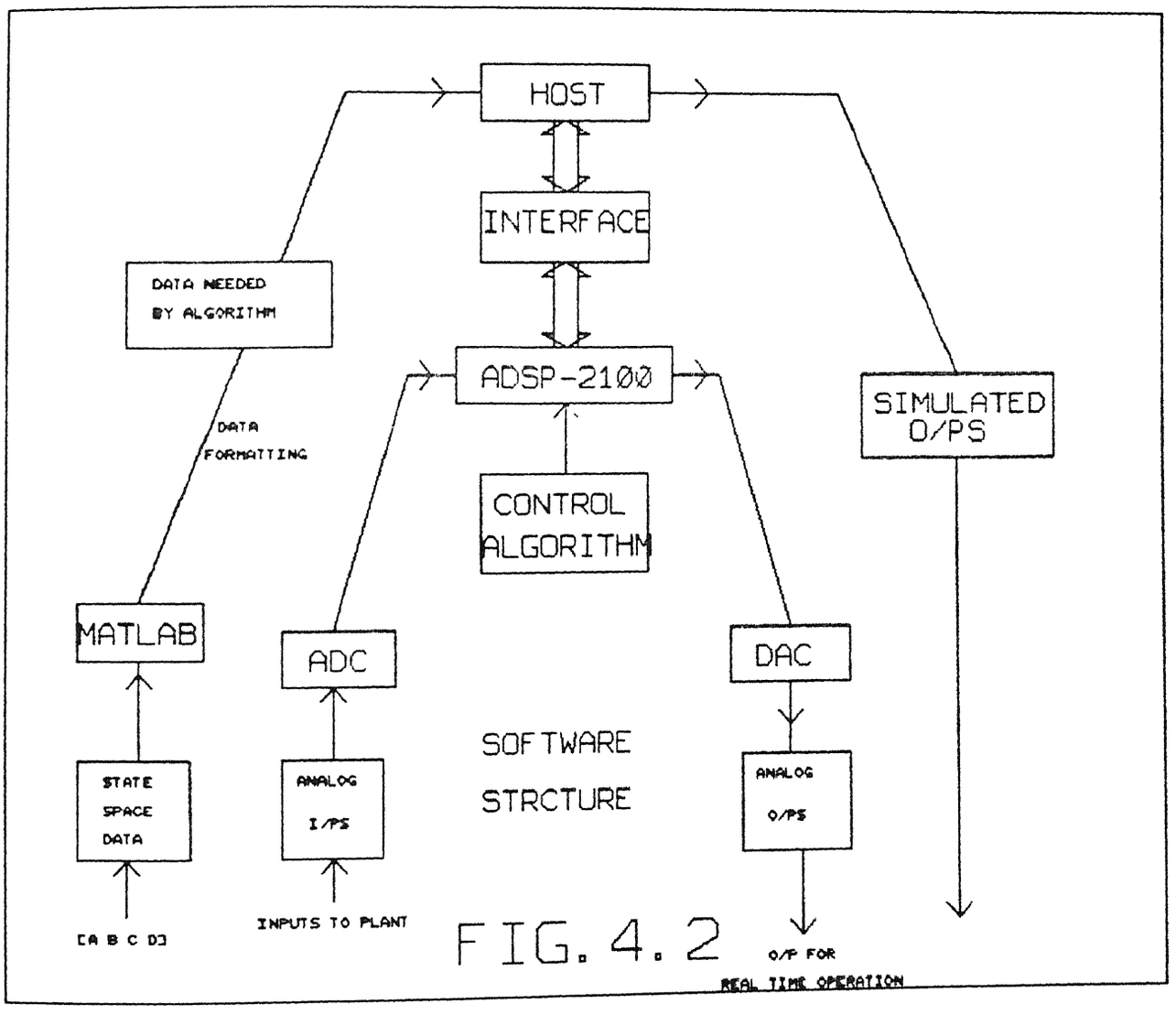

HOST
INTERFACE
DATA NEEDED BY ALGORITHM
ADSP-2100
SIMULATED O/PS
DATA FORMATTING
CONTROL ALGORITHM
MATLAB
ADC
DAC
STATE SPACE DATA
ANALOG I/PS
ANALOG O/PS
SOFTWARE STRCTURE
[A B C D]
INPUTS TO PLANT
O/P FOR REAL TIME OPERATION

FIG. 4.2

which are known [Appendix B].The MATLAB thus does the data formatting.

The algorithm before being subjected to real-time operation is simulated for identifying the program bugs and evaluating the system performance.Once this is done, the inputs to the plant are given to the processor and the outputs are available for real-time operation .

4.1.3 The offline compuational controller algorithm --

Let the discrete LTI system be given by

$$x(k+1) = Ax(k) + Bu(k)$$
$$y(k) = Cx(k) + Du(k) \qquad \text{-------(1)}$$

which is of order n and r-input,t-output.

Let the system described by eqn. [1] be denoted by P=[A B C D] (n-th rxt) where A, B, C, D are the matrices used in eqn.(1). Similarly the state-space realization of r.c.f's of the plant and the controller Np, Dp, X, and Y is known.[appendix B] For that follow the steps given below.

Step 1-- Select constant matrices K and F such that A0=A-BK and A1=A-FC are stable ie have poles in the closed unit disk, using pole placement.

step 2-- Find out the state-space realization for r.c.f.of the plant (Np,Dp) [equation (i) and (ii) - Appendix B]

$$Np = [A0 \; B \; C-DK \; D]$$
$$Dp = [A0 \; B \; -K \; I]$$

$$Dp^{-1} = [A\ B\ \ K\ I]$$

Step 3-- Find the state-space realization for X and Y satisfying the Bezout identity YDp+XNp=I [equation (v) and (vi) - Appendix B]

$$X = [A1\ F\ K\ 0]$$

$$Y = [A1\ B-FD\ K\ I]$$

Step 4-- Choose any stable matrix R = [Ar Br Cr Dr] of dimensions Ar(nxn), Br(nxr), Cr(txn), and Dr(txr).

The steps followed for developing the the software are given here. Refer fig.4. y(i) and u(i) respectively denotes the discrete outputs and inputs at smaple time t=i.

(I) - Given the inputs to the plant u(i), find u1(i) = u(i) - y5(i-1)

(II) - Dp^{-1}=[A1 B1 C1 D1], with input as u1(i) get the corresponding y1(i).

(III) - Np=[A2 B2 C2 D2], with y1(i) as input get the output y2(i). y2(i) is the output of the controlled plant.

(IV) - X=[A3 B3 C3 D3], find out y3(i) with y2(i) as the input.

(V) - Y=[A4 B4 C4 D4], find y4(i) with u1(i) as input.

(VI) - u5(i) = y3(i) + y4(i)

(VII) - R=[Ar Br Cr Dr] , find y5(i) with u5(i) as input.

(VIII) - Get the next sample of input sequence u(i+1) at t=i+1. Go to step (I) and repeat.

The software was entirely written in assembly language.Whereas this has a disadvantage in the prolonged development time,performance in terms of execution speed and memory usage is improved.The ADSP-2100 is supported with a set of software and hardware

development tools.The software was designed using the ADSP system cross software development tools and the Evaluation board is used for real time opeartion.

SECTION 4.2 -- SYSTEM DESIGN

The feedback controller algorithm developed above has been tested on the evaluation board described in chapter 3 [section 3.3.2]. But for the controller to be used in industry, a following design is suggested. It consists of an ADSP-2100 system, an I/O board for handling both analog as well as digital signals, and a interface to the IBM PC/AT for storing as well as transferring the plant data required by the ADSP system.

The functional block diagram of the complete system is shown in fig.4.3. Each of the blocks is described below.

(i) ADSP-2100 system with PM and DM --

The ADSP-2100 takes a TTL-compatible clock signal CLKIN, running at four times the processor cycle time as an input.Using this clock input, the processor divides the internal processor cycle into eight states, defined by the edges of the input clock.The active processor cycle consists of state 1 through 7. State 8 is dead-zone to provide a neutral stopping point for halting the processor.

The processor interfaces with the program memory with PMA bus for addressing and PMD for fetching the instructions. As PM is used for data as well as instruction PMDA signal is

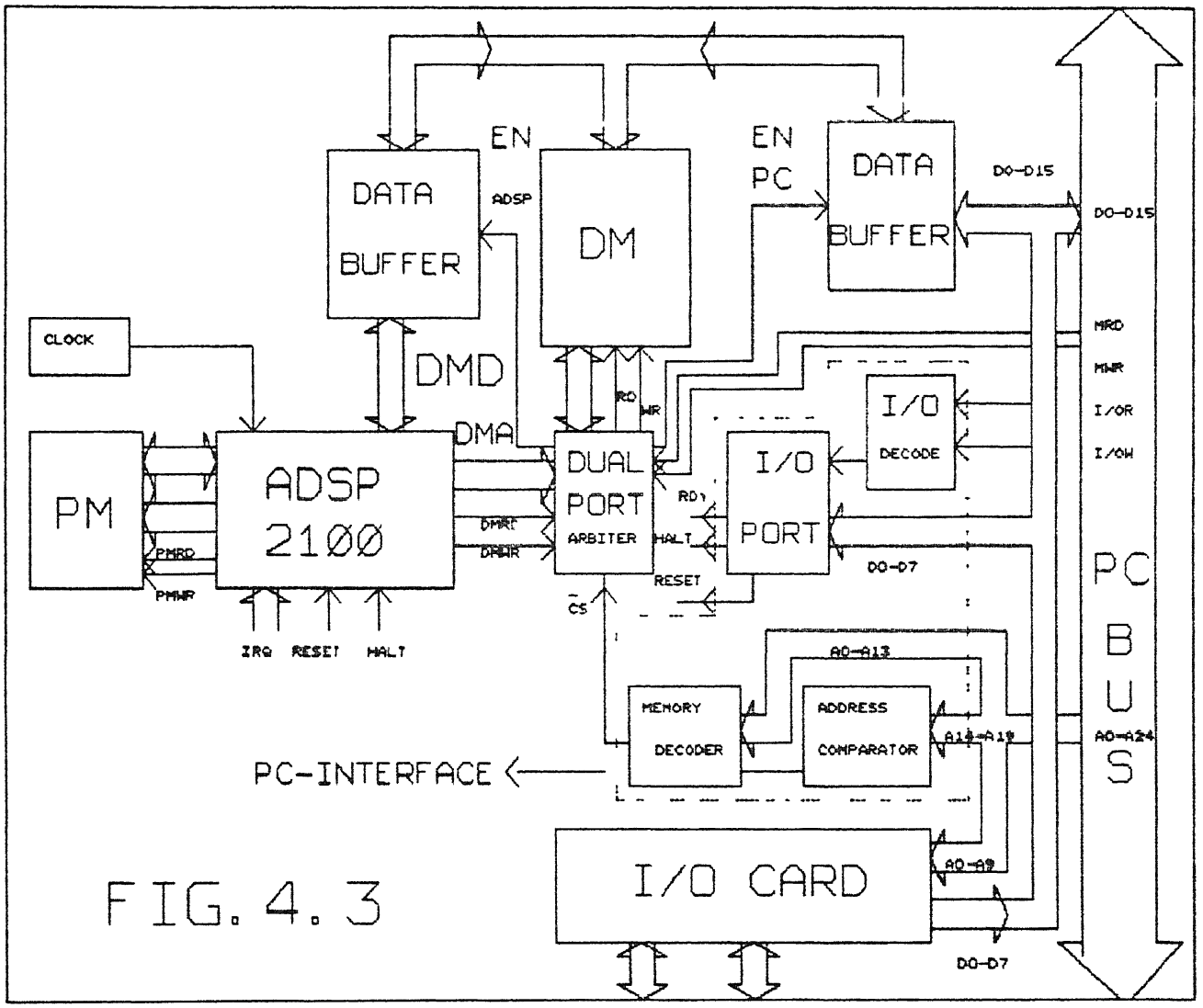

EN
EN
PC
DATA
BUFFER
ADSP
DM
DATA
BUFFER
D0-D15
D0-D15
CLOCK
DMD
MRD
MWR
I/OR
I/OW
RD
WR
I/O
DECODE
DMA
PM
ADSP
2100
DUAL
PORT
ARBITER
I/O
PORT
RD
HALT
RESET
D0-D7
PC
B
U
S
PMRD
PMWR
IRQ
RESET
HALT
A0-A13
MEMORY
DECODER
ADDRESS
COMPARATOR
A0-A24
PC-INTERFACE
I/O CARD
A0-A9
D0-D7

FIG. 4.3

asserted whenever data instead of instruction is fetched.PMS, PMWR, PMRD could be used as chip select,write and read strobe signals respectively.

The ADSP interfaces with the data memory (DM) through a dual port arbiter.The DM is also shared by the PC bus for required data transactions. Whenever the En-ADSP signal is active, the ADSP can access DM. The processor as well as the the host PC can also write to and read from the DM.

(ii) PC BUS --

The I/O channel of the PC/AT supports I/O address space at 100 to 3FF hex.The bus contains 24-bit memory address lines, 16/8 bit data bus, interrupts, control lines for memory and I/O read and write lines etc.Some of these signals which are used are described here.

SA0-SA19(I/O) --

Address bits through 0 to 19 are used to address memory and I/O within the system.These signals are gated on the system bus when the 'BALE' signal is high and are latched on the falling edge of 'BALE'.

SD0-SD15(I/O)--

These signals provide bits 0 to 15 for the processor of PC, memory and I/O devices.D0 is the least significant bit.All the 8-bit devices should use D0-D7 for communication to the microprocessor.The 16-bit device will use all the D0-D15 lines.

-IOR(I/O) --

This instructs an I/O device to drive its data onto the data bus.It may be driven by the system microprocessor or DMA

controller, or by another processor resident on the I/O channel.

-IOW(I/O)--

This instructs an I/O device to read the data on the data bus. It may be driven by any microprocessor in the system or resident in the I/O channel.

(iii) Dual port arbiter--

The arbiter serves to share a resource onto a multi-master system and avoid the contention problem between various masters. The arbiter shown in fig.4.3 shares data memory between two devices, ADSP-2100 and the PC/AT bus.Priorities are assigned to both ADSP and PC, the higher one being assigned to ADSP. Depending on the requests for the DM and considering the priority, the arbiter generates En-ADSP or En-PC. AS soon as the data is available in the DM, the ADSP takes over the DM and starts functioning. After it is finished off, it releases the control of DM and allows the PC to transfer next data.

(iv) PC interface unit--

This unit interfaces the PC bus with the rest of the ADSP system. The interface unit provides a port on which a PC can write and from which it can read. The 16k block of DM is mapped within the memory space of PC. The unit therefore has a memory decoder logic to decode the block of DM. In addition it provides signals which can interrupt/reset the ADSP processor as well as be interrupted and halted by the ADSP processor.

(v) I/O card --

This board has capabilities for analog as well as digital inputs and outputs. The RTI-815 is such a IBM PC/XT/AT compati-

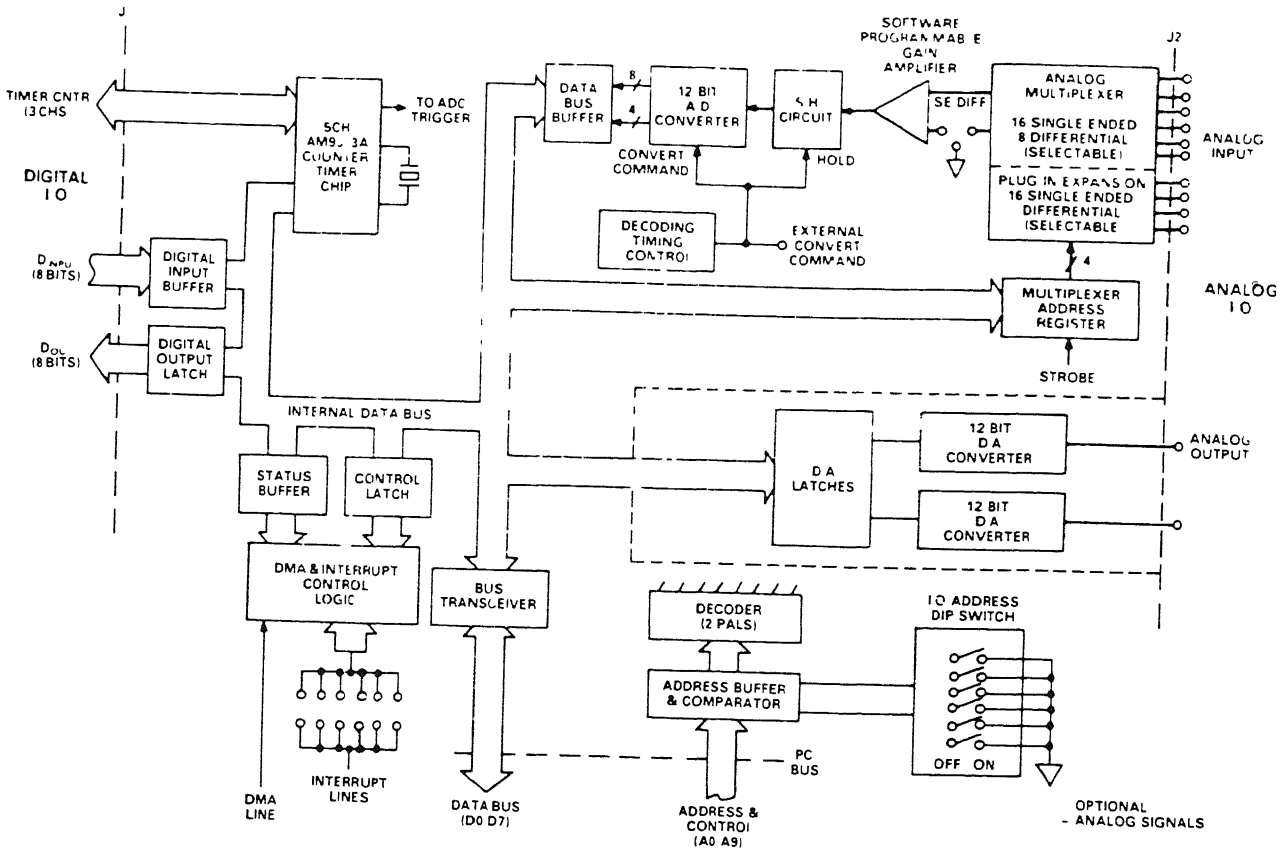


FIG. 4.4

RTI-815 Block Diagram

ble board from analog devices [12]. The block diagram of this is shown in fig.4.4 .There are two input multiplexers providing capability for 16-single ended inputs. Two analog output channels are provided.The digital input/output port take 8-bit parallel latching inputs and non-latching outputs.

The interrupt generator is compatible with the interrupts request lines of IBM PC. Along with DMA, data can be transferred using a burst mode.

SECTION 4.3 -- A CASE STUDY

A discrete LTI system (4-th order,2-input 1-output) is taken as an example.

Let the system be described by P=[A B C D] where

$$A = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \qquad B = \begin{bmatrix} 1 & 2 \\ 0 & 2 \\ 0 & 2 \\ 0 & 2 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \qquad D = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

Select any K and F such that the poles of A-BK and A-FC are within the closed unit disc.

$$K = \begin{bmatrix} -4.2 & 6.3 & -3.9 & 0.8 \\ 0.5 & 0.9 & 0.5 & -0.3 \end{bmatrix}$$

$$F = \begin{bmatrix} 1.3 \\ 5.5 \\ 8.0 \\ -4.75 \end{bmatrix}$$

The state-space realization of Dp^{-1} , Np , X , and Y are as follows.

Dp^{-1} = [Ad Bd Cd Dd]

$$Ad = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \qquad Bd = \begin{bmatrix} 1 & 2 \\ 0 & 2 \\ 0 & 2 \\ 0 & 2 \end{bmatrix}$$

$$Cd = \begin{bmatrix} -4.2 & 6.3 & -3.9 & 0.8 \\ 0.5 & 0.9 & 0.5 & -0.3 \end{bmatrix} \qquad Dd = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

,

Np = [An Bn Cn Dn] where,

$$An = \begin{bmatrix} -2.2 & -4.4 & 2.9 & -0.2 \\ 0.0 & 0.9 & -1 & 0.6 \\ -1 & 2.9 & -2 & 0.6 \\ -1 & 1.9 & 0.0 & -0.3 \end{bmatrix} \qquad Bn = \begin{bmatrix} 1 & 2 \\ 0 & 2 \\ 0 & 2 \\ 0 & 2 \end{bmatrix}$$

$$Cn = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \qquad Dn = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

X = [Ax Bx Cx Dx] where

$$Ax = \begin{bmatrix} -1 & 0 & 0 & -1.3 \\ 1 & -1 & 0 & 5.5 \\ 0 & 1 & -1 & -8 \\ 0 & 0 & 0 & 3.7 \end{bmatrix} \qquad Bx = \begin{bmatrix} 1.3 \\ 5.5 \\ 8 \\ -4.75 \end{bmatrix}$$

$$Cx = \begin{bmatrix} -4.2 & 6.3 & -3.9 & 0.8 \\ 0.5 & 0.9 & 0.5 & -0.3 \end{bmatrix} \qquad Dx = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

Y = [Ay By Cy Dy] where

$$Ay=\begin{bmatrix} -1 & 0 & 0 & -1.3 \\ 1 & -1 & 0 & 5.5 \\ 0 & 1 & -1 & -8 \\ 0 & 0 & 0 & 3.7 \end{bmatrix} \quad By=\begin{bmatrix} 1 & 2 \\ 0 & 2 \\ 0 & 2 \\ 0 & 2 \end{bmatrix}$$

$$Cy=\begin{bmatrix} -4.2 & 6.3 & -3.9 & 0.8 \\ 0.5 & 0.9 & 0.5 & -0.3 \end{bmatrix} \quad Dy=\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

,

R = [Ar Br Cr Dr] which is specified independent of the plant model.

$$Ar=\begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix} \quad Br=\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$Cr=\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad Dr=\begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

given an input sequence, the simulated output can be seen on the simulator.

As shown in the case study,all the controller parameters except R are determined by the plant model.The R part is independent and hence could be varied to meet different requirements.Essentially there is no limit on the complexity of the controller which could be implemented and is achieved simply by changing the constants (system order and no.of inputs/outputs) in the program.The following tables show the controller complexity Vs execution time.

Table 1 shows system order Vs execution time for a 2-input 2-output system.Table 2 shows the system complexity

ORDER	EXECUTION TIME MICROSEC
4	244
5	307
6	385
7	474
8	573
9	682
10	800

TABLE 1

NO OF I/O	EXECUTION TIME MICROSEC
1X1	218
2X2	244
3X3	417
4X4	542

TABLE 2

OPDER(I/O)	PM CODE (WORDS)	PM DATA (WORDS)	DM (WORDS)
4(4X4)	450	64	288
4(2X2)	450	36	288
4(3X4)	450	56	288
5(2X2)	450	49	288
6(2X2)	450	64	288
7(2X2)	450	81	288
8(2X2)	450	100	288
9(2X2)	450	121	288
10(2X2)	450	144	288

TABLE 3

ın terms of no. of ınputs and outputs Vs execution tıme.Table 3 shows the execution time as the complexıty of the system is increased in terms of no.of ınputs and outputs as well as order of the system.

CHAPTER 5

DESIGN CONCLUSION

SECTION 5.1 -- CONCLUSION

As shown in the fig.1.2 [Chapter 1], the stabilizing controller has a free part which could be any stable transfer function.If in addition to close loop stability,a close loop performance is also expected,only the free part has to satisfy some constraint.In either case the fixed part remains unchanged.So for obtaining different performance requirements from the closed loop system,one need to chose the free part appropriately.This is the biggest advantage.If the plant happened to be perturbed slightly from its equilibrium position,there is no need to reformulate the stabilization problem. Hence the retuning of the controller parameter is avoided. The free part alone determines the closed loop performance hence only it can be changed to refine the performance.

The powerful architecture of the ADSP-2100 lends itself to meet different requirements in the implementation.It supports interfacing peripherals often needed in the control loop. In addition,from the tables shown in section 4.3 they are advantageous in terms of speed.

In conclusion, it can be said that, although the factorization theoretic controllers algorithm is more complex than the traditional control algorithms, it can be implemented using the commonly available hardware for simple cases of practical interest.

A superior control system performance can be expected from such implementations.

SECTION 5.2 -- SCOPE FOR FURTHER WORK

This thesis has demonstrated the efficiency,superiority as well as practical feasibility of factorization theoretic controllers; both from the point of view of algorithm and implementation.This theory is used to solve several important control problems.So the next task would be to use these controller formulae to develop efficient algorithms which can be implemented using the available hardware technology.

There has been an explosive growth in late 80's in the signal processing devices.With ready availability of micron and submicron technology DSP chips with clock rates in excess of 25MHz are available. With excellent I/O interfacing capabilities,they are tending to be more reliable.Thus faster and reliable ways of implementation could easily be achieved.Integration of these modern devices and efficient algorithms would lead to a high performance control for variety of application.

APPENDIX -A

Let P be any discrete LTI Plant and let it be a proper rational Matrix. A right coprime factorization (r.c.f) of P is $P=NpDp^{-1}$,where Np and Dp are stable and right coprime transfer functions. Similarly left coprime factorization of P has a form $P=\bar{D}p^{-1}\ \bar{N}p$, where $\bar{N}p$ and$\bar{D}p$ are stable and left coprime transfer functions.

The_stabilization_problem -- [1]

The purpose here is to stabilize P by C which could be parametrized via a parameter R which is any stable transfer matrix ie to find a formula for the controller C so that all LTI discrete time controllers which provide closed loop stability are of the some form.

Corresponding to the rcf and lcf ,there exists four more matrices satisfying the following

$$P= NpDp^{-1} = \bar{D}p^{-1}\ \bar{N}p$$

$$\begin{bmatrix} Y & X \\ -\bar{N}p & \bar{D}p \end{bmatrix} \begin{bmatrix} Dp & -\bar{X} \\ Np & \bar{Y} \end{bmatrix} = I \qquad \text{--------- (1)}$$

Eq.(1) is called as doubly coprime factorization. P is said to be stabilizable if there exists a proper rational C which stabilizes it. The pair (P,C) is said to be stable. In this view the following lemma is of importance .

Let the coprime factorization of C is $C = \bar{U}\ \bar{V}^{-1} = V^{-1}U$

Lemma_1 :

If C stabilizes P then the following statements are equivalent

(i) C stabilizes P .

(ii) $\begin{bmatrix} Dp & -\bar{U} \\ Np & \bar{V} \end{bmatrix}^{-1}$ is a stable transfer function.

(iii) $\begin{bmatrix} V & U \\ -\bar{N}p & \bar{D}p \end{bmatrix}^{-1}$ is a stable transfer function

Parameterization :

Thoerem : The set of all C's stabilizing P are parametrized as

$$C = (\bar{X} + DpR) (\bar{Y} - NpR)^{-1} \qquad \text{----------(2)}$$

$$= (Y - R\bar{N}p)^{-1} (X + R\bar{D}p) \qquad \text{----------(3)}$$

Proof :

Part(1) : First we will show that if C is given by Eq.(2), then it stabilizes P.

Define

$$U = (X + R\bar{D}p) \qquad V = (Y - RN\bar{p})$$

$$\bar{U} = (\bar{X} + DpR) \qquad \bar{V} = (\bar{Y} - NpR)$$

Consider,

$$\begin{bmatrix} V & U \\ -\bar{N}p & \bar{D}p \end{bmatrix} \begin{bmatrix} Dp & -\bar{U} \\ Np & \bar{V} \end{bmatrix}$$

$$= \begin{bmatrix} (Y - R\bar{N}p) & (X + R\bar{D}p) \\ -\bar{N}p & \bar{D}p \end{bmatrix} \begin{bmatrix} Dp & -(\bar{X} + DpR) \\ Np & (\bar{Y} - NpR) \end{bmatrix}$$

$$= \begin{bmatrix} I & R \\ 0 & I \end{bmatrix} \begin{bmatrix} Y & X \\ -\bar{N}p & \bar{D}p \end{bmatrix} \begin{bmatrix} Dp & -\bar{X} \\ Np & \bar{Y} \end{bmatrix} \begin{bmatrix} I & -R \\ 0 & I \end{bmatrix}$$

$$= \begin{bmatrix} I & R \\ 0 & I \end{bmatrix} \begin{bmatrix} I & -R \\ 0 & I \end{bmatrix}$$

$$\begin{bmatrix} V & U \\ -\bar{N}p & \bar{D}p \end{bmatrix} \begin{bmatrix} Dp & -\bar{U} \\ Np & \bar{V} \end{bmatrix} = I \qquad \text{----------(4)}$$

Therefore $\bar{U}$, $\bar{V}$, are rcf and U, V are lcf of C. From eq.(4) it follows that

$$\begin{bmatrix} Dp & -\bar{U} \\ Np & \bar{V} \end{bmatrix}^{-1}$$ is a stable transfer function.

From Lemma(1), it follows that C stabilizes P.

Part(2) :

We now show that if C stabilizes P then C satisfies Eq.(2) for some stable transfer matrix R.

Consider

$$\begin{bmatrix} Y & X \\ -\bar{N}p & \bar{D}p \end{bmatrix} \begin{bmatrix} Dp & -\bar{U} \\ Np & \bar{V} \end{bmatrix} = \begin{bmatrix} I & (-Y\bar{U} + X\bar{V}) \\ 0 & (\bar{N}p\bar{U} + \bar{D}p\bar{V}) \end{bmatrix}$$

Let $\bar{N}p\bar{U} + \bar{D}p\bar{V} = D$

$$= \begin{bmatrix} I & (-Y\bar{U} + X\bar{V}) \\ 0 & D \end{bmatrix}$$

Choose $R = -(X\bar{V} - Y\bar{U}) D^{-1}$

$$\begin{bmatrix} Y & X \\ -\bar{N}p & \bar{D}p \end{bmatrix} \begin{bmatrix} Dp & -\bar{U} \\ Np & \bar{V} \end{bmatrix} = \begin{bmatrix} I & -RD \\ 0 & D \end{bmatrix}$$

Premultiply both sides by $\begin{bmatrix} Dp & -\bar{X} \\ Np & \bar{Y} \end{bmatrix}$

$$\begin{bmatrix} Dp & -\bar{U} \\ Np & \bar{V} \end{bmatrix} = \begin{bmatrix} Dp & -\bar{X} \\ Np & \bar{Y} \end{bmatrix} \begin{bmatrix} I & -RD \\ 0 & D \end{bmatrix}$$

$$= \begin{bmatrix} Dp & -(\bar{X} + DpR) \\ Np & (\bar{Y} - NpR) \end{bmatrix}$$

$$\bar{U} = (\bar{X} + DpR) D \quad \text{and}$$

$$\bar{V} = (\bar{Y} - NpR) D$$

$$C = \bar{U} \bar{V}^{-1} = (\bar{X} + DpR) (\bar{Y} - NpR)^{-1}$$

which is Eq.(2). Hence if C stabilizes P, C is given by Eq.(2) and Eq.(3) [It can be similarly shown that C satisfies Eq.(3)], for any R which is a stable transfer function matrix.

APPENDIX B

State space formulae for doubly coprime factorization for a discret time system--

A method for obtaining left and right coprim factorization of a system from its state space description is presente here.As these formulae use state space realization, they can be easil implemented using a computer.

Consider the discrete model of a LTI plant given by

$$x(k+1) = A\,x(k) + B\,u(k) \qquad \text{----------(1)}$$

$$y(k) = C\,x(k) + D\,u(k)$$

Let the transfer function matrix $C(zI-A)^{-1}B+D$ is denoted by $[A\ B\ C\ D]$.The doubly coprime factorization of P is given by

$$\begin{bmatrix} Y & X \\ -\bar{N}p & \bar{D}p \end{bmatrix} \begin{bmatrix} Dp & -\bar{X} \\ Np & \bar{Y} \end{bmatrix} = I \qquad \text{--------(2)}$$

and can be derived as follows.
We have $P = D+C(zI-A)^{-1}B = [A\ B\ C\ D]$ and given that (A,B) and (A,(are stabilizable and detectable.

Let U be any close superset of the complement the open unit disk which is symmetric with respect to the real-axis.N choose a real matrix K such that A0=A-BK is stable ie all the eigenva ues are contained in C-U where C is the entire z-plane.

Define $v(k) = u(k) + K\,x(k)$

From equation (1) we get

$$x(k+1) = Ax(k)+Bv(k)-BKx(k) = (A-BK)x(k) + Bv(k)$$

$$u(k) = -Kx(k) - v(k)$$

$$y(k) = Cx(k) + Dv(k) - DKx(k)$$

$$= (C-DK)x(k) + Dv(k)$$

The transfer matrix from v(k) to u(k) and from v(k) to y(k) are

$u(k)=Mv(k)$, $y(k)=Nv(k)$;

so $y(k) = NM^{-1} u(k)$

ie $P = NM^{-1}$

so M=Dp and N=Np and are therefore given by

$Dp=[A-BK\ B\ -K\ I] = I-K(zI-A0)^{-1}B$ ----------(1)

similarly

$Np=[A-BK\ B\ C-DK\ D] = D+(C-DK)(zI-A0)^{-1}B$ --------(ii)

Now choose F such that A1=A-FC is stable.Then following similar steps as above we get state-space realization for the left coprime factorization $\overline{Dp},\overline{Np}$; $P= \overline{Dp}^{-1}\overline{Np}$

$\overline{Dp}=[A1\ F\ -C\ I] = I-C(zI-A1)^{-1}F$ ----------(iii)

$\overline{Np}=[A1\ B-FD\ C\ D] = D+C(zI-A1)^{-1}(B-FD)$ --------(iv)

Now let's find $X,Y,\overline{X},\overline{Y}$ in equation (2)

From appendix-A we know that if C stabilizes P iff C is of the form $C=Y^{-1}X$ where X and Y satisfy YDp+XNp=I

First we will find C which will stabilize P by the observer theory.The state-space equation for such is given by

$$\hat{x}(k+1)=A\hat{x}(k)+Bu(k)-F((C\hat{x}(k)+Du(k))-y(k)) \quad ---(3)$$

$$u(k) = K\hat{x}(k)$$

or

$$\hat{x}(k+1)=(A+BK-FC-FDK)\hat{x}(k)+Fy(k)$$

$$u(k)=K\hat{x}(k)$$

Let $\hat{A}=A+BK-FC-FDK$

$\hat{B}=F$

$\hat{C}=K$ ---(4)

therefore

$$\hat{x}(k+1)=\hat{A}\hat{x}(k)+\hat{B}y(k)$$

$$u(k)=\hat{C}\hat{x}(k)$$

From equation(4) the controller $C=[\hat{A}\ \hat{B}\ \hat{C}\ 0]$

$$C=Y^{-1}X$$

Let $\hat{F}$ be a matrix such that $\hat{A}-\hat{F}\hat{C}$ be stable

Choose $\hat{F}=B-FD$ so,

$$\hat{A}1=\hat{A}-\hat{F}\hat{C} = A+BK-FC-FDK-(B-FD)K$$

$$=A-FC = A1$$

Using equation (iii) and (iv),we get

$$Y=[\hat{A}1\ \hat{F}\ -\hat{C}\ I]$$

$$=[A1\ B-FD\ -K\ I] = I-K(zI-A1)^{-1}(B-FD) \quad \text{----------(v)}$$

$$X=[\hat{A}1\ \hat{B}-\hat{F}\hat{D}\ \hat{C}\ \hat{D}]$$

$$=[A1\ F\ K\ 0] = I-K(zI-A1)^{-1}F \quad \text{------------(vi)}$$

A similar procedure leads to the r.c.f of $C=XY^{-1}$ where

$$\bar{Y}=[AO\ F\ C-DK\ I] = (C-DK)(zI-AO)^{-1}F+I \quad \text{------------(vii)}$$

$$\bar{X}=[AO\ F\ K\ 0] = K(zI-AO)^{-1}F \quad \text{--------------(viii)}$$

EE-1994-M-KUL-PTA